

3 Un año de emprendimiento

Revista de Software Libre ATIX 2009

Equipo Editorial

Colaboradores

Dirección y Coordinación General

Esteban Saavedra López (jesaavedra@opentelematics.org)

Diseño y Maquetación

Jenny Saavedra López (jennysaavedra@gmail.com)

Esteban Saavedra López (jesaavedra@opentelematics.org)

Revisiones

Esteban Saavedra López

Jenny Saavedra López

Noticias

Ivonne Menacho

Jenny Saavedra López

Marcia Velasquez

Autores Frecuentes

Arnold Guzmán

Carlos Ramos

Ernesto Rico Smith

Esteban Saavedra López

Joseph Sandoval

Lucy Ayarde Romero

Rocio Figueroa

Williams Chorolque Choque

Herramientas

La edición de esta revista fue realizada de forma integra haciendo uso de Software Libre









Palabra quechua,

con un sentimiento profundo

y con gran significado filosófico

El que lo sabe

El que lo intenta

El que lo puede

El que lo logra

El que lo sabe, el que lo intenta, el que lo puede, el que lo logra

Editorial

Al empezar un nuevo proyecto o una nueva actividad en nuestras vidas, siempre sentimos el miedo de saber como nos irá, pero al completar un ciclo, nos damos cuenta realmente de lo difícil o sencillo que fue, de las cosas que olvidamos, de las cosas que aprendimos, de las cosas que faltaron; lo importante es que aprendimos, y ganamos experiencia, aspectos que nos ayudaran a encarar un nuevo ciclo en nuestras vidas, en nuestro conocimiento y en nuestros proyectos

Un año de emprendimiento, un título que refleja de forma clara y concisa lo que el equipo de la revista Atix trato de hacer; hace un año decidimos emprender este reto, que sabíamos que era una tarea difícil, con muchísimas falencias presupuestarias y logísticas, pero a cambio proponíamos muchas ganas, mucho esfuerzo, y sobre todo mucha predisposición a salir adelante, ahora después de haber concluido un año, nos damos cuenta que nuestro compromiso dió sus frutos, no solo para quienes conforman el equipo de la revista, sino también para nuestros lectores, y autores, por que estamos seguros que lo logrado en este año de emprendimiento vale más que cualquier sacrificio y restricción que hayamos podido tener.

En éste noveno número ponderamos los mensajes de libertad y aliento que recibimos de la comunidad nacional e internacional, mensajes que nos motivan y comprometen más a seguir trabajando en bien del conocimiento, las tecnologías y el software libre, por que esto representa nuestra libertad.

Un año de emprendimiento y la satisfacción de haber cumplido nuestro primer ciclo.

Bienvenidos a nuestro noveno número

Esteban Saavedra López
Director y Coordinador General

Contenido

Liberado el 20 de abril de 2009



Programando con NCURSES en C

Ncurses es una biblioteca de programación que provee una API que permite al programador escribir interfaces basadas en texto, TUIs. También optimiza el refresco de la pantalla, lo que permite reducir la latencia experimentada cuando se usan intérpretes de comandos remotos. Incluso en las interfaces de usuario modo texto podemos elaborar aplicaciones elegantes y llamativas.



Introducción

En estos días en donde las interfaces gráficas dominan la mayor parte de las aplicaciones que se desarrollan, algún lector podría pensar que escribir un artículo de programación sobre interfaces en modo texto (TUIs) es algo anticuado, sin embargo, es cierto que hasta hoy en día aún se emplean este tipo de programas, en unos sistemas un poco más que en otros. Dentro de los sistemas tipos UNIX aún es muy común interactuar con estos programas y los sistemas GNU/Linux no podrían ser la excepción. Usted como usuario de estos sistemas o como usuario normal lo más probable es que en algún momento se halla enfrentado con uno de estos programas; si usted es un usuario acostumbrado a las interfaces gráficas de usuario (GUIs) quizás no le fue de mucho agrado trabajar con estos programas.

Hace apenas unas 3 décadas (fácil decirlo) no existían las interfaces gráficas, todo programa estaba orientado a la consola. Poco a poco se buscaron las formas para hacer que estos programas fuesen más intuitivos y amigables para el usuario; en afán de alcanzar este propósito se crearon APIs para elaborar aplicaciones de usuario, con elementos fáciles de identificar y manejar, por ejemplo, colores, menús, uso del ratón, etc. Una de estas APIs fue desarrollada por la compañía Borland, un conjunto de bibliotecas no estándar para lenguaje C que incluyó en su famosos IDEs Borland C y Turbo C. Quién no conoce las famosas **`conio.h`**, **`dos.h`**, **`graphics.h`**, etc. con las que podíamos crear incluso vídeo juegos. Pues bien, en este artículo aprenderemos de forma básica como hacer esto pero en sistemas GNU/Linux.

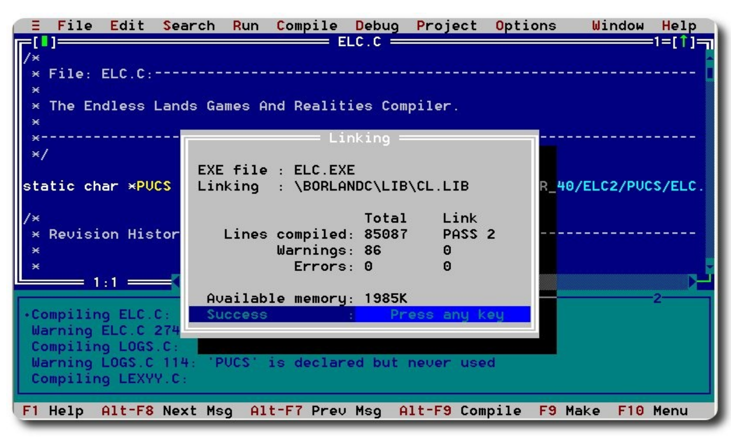


Gráfico 1.



Otro proyecto famoso dentro de los entornos UNIX fue CURSES, del cual hablaremos en la siguiente sección.

El presente documento tiene la finalidad de introducir al usuario a la teoría y algo de práctica con la biblioteca **ncurses**, no pretende ser un tutorial extenso y formal. Se espera que con la información que aquí se maneje se pueda dotar al usuario de las bases necesarias para poder investigar y mejorar por su parte. Al final del documento se listan algunas referencias y enlaces a material muy valioso que puede ser de utilidad para el lector.

¿Qué es NCURSES?

Antes de conocer ciertamente el significado de **ncurses** tenemos que hablar sobre Curses, una biblioteca creada por Ken Arnold lanzada inicialmente bajo un sistema UNIX BSD. Esta biblioteca contenía funciones para el manejo de interfaces en modo texto (TUIs), las cuales permitían la manipulación de varios elementos del terminal, tales como los colores, manejo del cursor, menús, etc. La biblioteca curses fue descontinuada hace algunos años.

Ncurses es el acrónimo de New Curses, uno de los proyectos de GNU bajo la licencia MIT, unos de los pocos que no está licenciado bajo las muy conocidas GPL o LGPL, licencias nativas del GNU. Hoy día, **Ncurses** viene a reemplazar esta biblioteca y ofrece todas las funciones antes mencionadas para todos los sistemas tipo-UNIX, incluso existe una versión para windows llamada PDCURSES.

¿Qué podemos hacer con ellas?

Con el uso de estas bibliotecas podemos elaborar aplicaciones en modo texto con interfaces llamativas y fáciles de usar. Podría contarles más pero como dice el dicho, una imagen dice más que mil palabras.

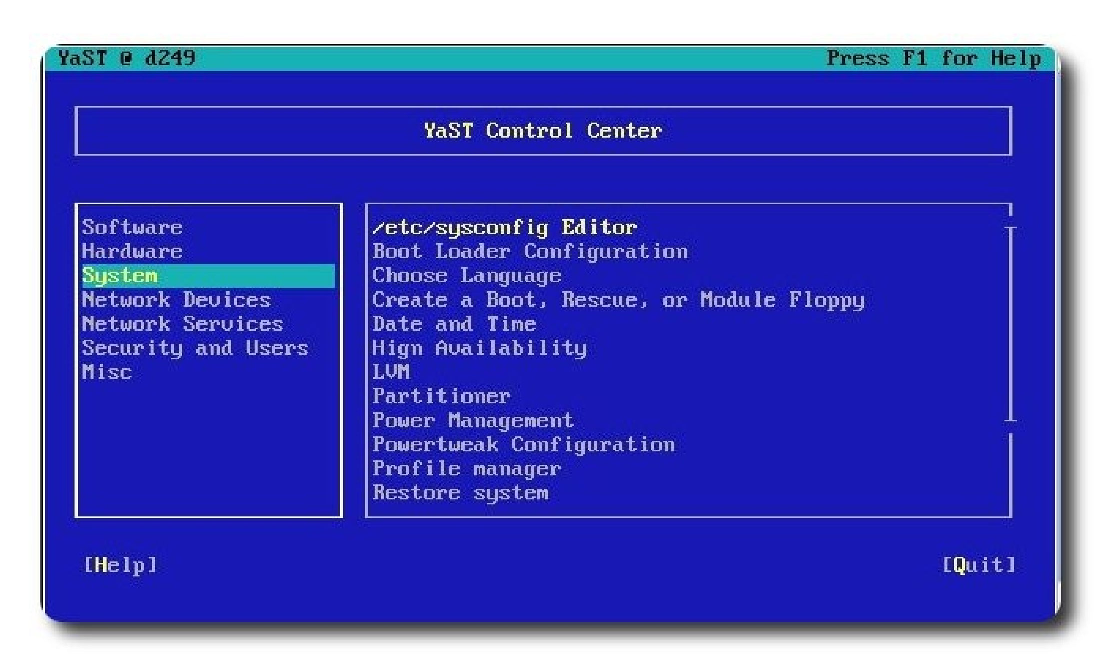


Gráfico 2.

En la imagen de arriba se muestra el programa Yast en modo consola que utiliza la distribución Suse para administrar el sistema. Además de este programa existen otros como por ejemplo, el programa para configurar la compilación del kernel, el cual ejecutamos con el comando **`make menuconfig`**.

Gráfico 3.

Creo que con éstos ejemplos es suficiente para darnos idea de que tipo de aplicaciones podemos elaborar con **Ncurses**.

¿En dónde las podemos utilizar?

Como ya se comentaba anteriormente, las aplicaciones basadas en interfaces modo texto a pesar de que aún se utilizan son ya muy pocas, con decir, que en sistemas como windows son raros los usuarios nuevos que saben de la existencia de este tipo de programas. Pero aplicándolo a nuestros sistemas GNU/Linux aún podemos elaborar y dar uso a éste tipo de aplicaciones, en el gráfico 2 se muestra un ejemplo del tipo de aplicaciones que se pueden realizar con **Ncurses**. Todo es cuestión de imaginación. Los programas que siempre hemos elaborado para visualizarse en la pantalla negra ahora pueden ser más amigables y elegantes si utilizamos las funciones de la biblioteca **Ncurses** para modelar la presentación del programa.

Sin más preámbulo pasemos ahora a trabajar con la biblioteca **Ncurses**, primero algo de teoría y después la práctica.

Instalación

Antes de iniciar con el tutorial es necesario preparar las herramientas necesarias para trabajar con **Ncurses**, las cosas que necesitaremos son el compilador de C (gcc) y por supuesto las bibliotecas **Ncurses**. Detallar la instalación abarcaría mucho espacio, por lo que saltaré este punto, además, la mayoría de las distribuciones tienen gestores de paquetes con los cuales será fácil instalar estas herramientas. O si lo prefieren pueden ir a la página oficial del proyecto en **`http://www.gnu.org/software/ncurses/`** y conseguir el material necesario.

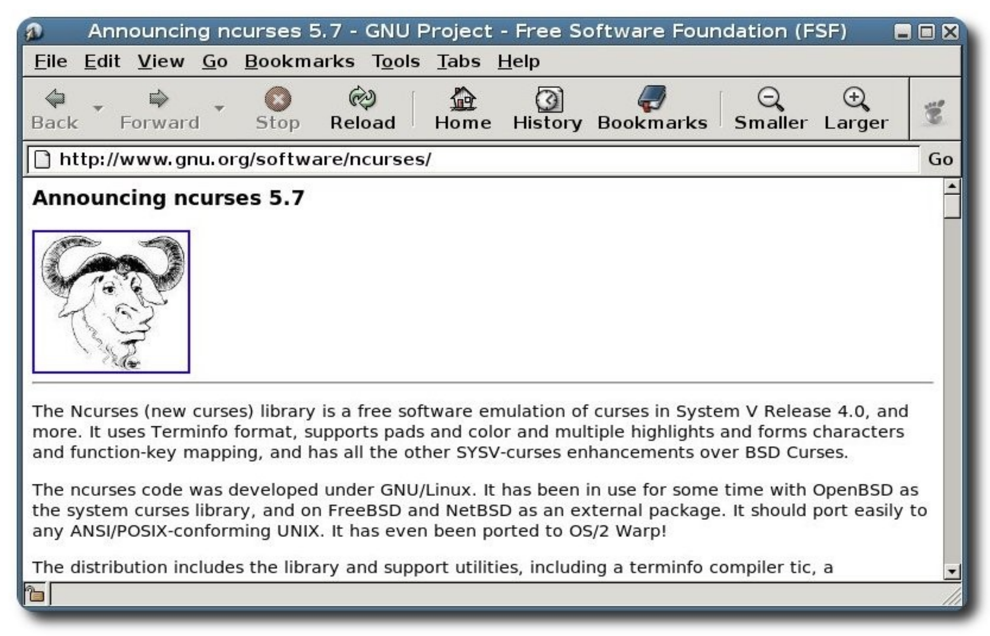


Gráfico 4: Página oficial del proyecto.

Empezando con NCURSES

Ventanas

Las ventanas son áreas rectangulares de n filas por m columnas sobre las cuales trabajaremos, realmente una ventana es una matriz de n*m en donde cada índice de la matriz es un espacio en el que se pueden escribir caracteres, ya sean símbolos o espacios en blanco en el caso de no tener nada en la pantalla. Esta comparación nos permite asimilar de mejor manera las operaciones sobre ventanas, tales como escribir caracteres en ciertas posiciones de pantalla en las cuales especificamos número de fila y número de columna, tal y como accedemos al índice de una matriz. Más adelante trataremos las operaciones sobre ventanas.

Por cierto, después de realizar operaciones en la pantalla debemos refrescarla (actualizar), llamar a una función para que haga visibles los cambios.

```
int refresh(void);
int wrefresh(WINDOW *win);
```

La primer función no recibe parámetros y refresca a la ventana estándar. La segunda hace lo mismo pero para una ventana en especifico, la cual se indica en su primer parámetro.

Cursores

Hay 2 tipos de cursores que debemos conocer, cursores físicos y cursores lógicos, los primeros hacen referencia al cursor que conocemos normalmente como puntero, el que manejamos con el mouse, por otra parte los cursores lógicos los cuales son conocidos como tal, los que señalan la posición de pantalla en la que se ha escrito el último carácter. Es necesario definir bien cada uno de ellos ya que realizan tareas diferentes, usaremos los cursores físicos con las funciones que nos brinda **Ncurses** para el manejo del mouse y los cursores lógicos para escritura de caracteres en pantalla.

Estructura de un programa

La estructura de un programa que utiliza las funciones de **Ncurses** es un poco diferente de un programa normal escrito en C, en nuestro caso en particular. Un programa escrito en C que utilice las funciones de la biblioteca **Ncurses** tiene la siguiente estructura:

- ✔ Código C antes del modo gráfico

- ✔ Inicialización del modo gráfico
- ✔ Uso de las funciones **Ncurses**
- ✔ Finalización del modo gráfico
- ✔ Código C después del modo gráfico

La estructura antes mencionada no es algo formal ni nada por el estilo, simplemente es la estructura que me parece más adecuada. A continuación se detalla cada una de estas partes.

Inicialización (initscr)

Como cualquier programa escrito en C, uno que utilice las funciones gráficas puede igualmente utilizar todas las funciones inherentes al lenguaje, ésta es la parte del código C antes del modo gráfico, en la cual podemos utilizar las funciones del lenguaje como queramos. En el momento que deseamos comenzar a trabajar con el modo gráfico y por ende la utilización de las funciones **Ncurses** debemos inicializar esta modalidad con la función **`initscr()`**, la cual preparará todo lo necesario para trabajar en modo gráfico.

Contenido

Después de iniciar el modo gráfico podemos hacer uso de todas las funciones que nos brinda la biblioteca, en esta área del programa podemos escribir leer y escribir datos sobre la pantalla, utilizar colores, uso del cursor, utilizar menús, etc.

Finalización

Cuando terminemos de utilizar el modo gráfico debemos llamar a la función **`endwin()`** para que restablezca el modo consola y se encargue de liberar la memoria utilizada, por poner un ejemplo. Después de esto es posible trabajar normalmente con el lenguaje, en este caso, C.

Funciones:

Inicializar

En seguida detallo algunas de las funciones más importantes para empezar a trabajar con **Ncurses**, no se entrará en gran detalle debido a que implicaría el uso de mucho espacio. Más adelante veremos como obtener información detallada de cada una de éstas funciones de las páginas de los manuales que incluye la documentación de **Ncurses**.

Como ya se mencionó, utilizaremos la función **`initscr()`** para iniciar el modo gráfico. No hay gran ciencia así que solo la llamamos para empezar a trabajar.

Crear y eliminar de Ventanas

El primer paso antes de utilizar las funciones de la biblioteca tenemos que crear una ventana sobre la cual trabajar, por defecto, al iniciar el modo gráfico se crea una ventana la cual se llama **`stdscr`**, por lo que en un principio es posible trabajar sobre ella sin necesidad de crear otra. Hay dos conjuntos de funciones, unas afectan a la ventana estándar, la que se crea por defecto, y el otro conjunto se aplica a ventanas especificas, las que nosotros creamos. Es fácil distinguir estos 2 conjuntos ya que la mayoría de las funciones del segundo conjunto terminan con la letra w. Veamos pues las funciones.

```
WINDOW *newwin(int nlines, int ncols, int begin_y, int begin_x);
WINDOW *subwin(WINDOW *orig, int nlines, int ncols, int begin_y, int begin_x);
```

Estas dos funciones nos permiten crear ventanas y subventanas respectivamente. Los argumentos son simples, con la primera función son 4 parámetros, **`nlines`** y **`ncols`** indican el número de filas por el número de columnas, los 2 argumentos restantes sirven para precisar la posición donde se empieza a dibujar la pantalla. La función **`subwin()`** utiliza un parámetro más, el primer parámetro debe indicar la venta padre para la nueva subventana. Ambas funciones retornan un puntero de tipo WINDOW que apunta a la

ubicación de la nueva ventana.

Para eliminar una ventana usamos **delwin**. El parámetro es el apuntador a la ventana que se quiere eliminar.

```
int delwin(WINDOW *win);
```

Entrada y salida de datos

Ncurses proporciona sus propias funciones para la entrada y salida de datos, en si son casi las mismas de la biblioteca estándar de C con algunas modificaciones para adaptarlas al modo gráfico. A continuación las funciones estándar y sus correspondientes en **Ncurses**.

C	Ncurses
printf()	Printw() y wprintw()
scanf()	scanw() y wcanw()
putc()	addch() y waddch()
getc()	getch() y wgetch()

Existen más funciones pero éstas son las más importantes. La única diferencia entre las funciones **Ncurses** que inician con la letra w, y las que no es que las primeras realizan operaciones sobre ventanas especificas, y las segundas las hacen sobre la ventana estándar **stdscr**, por lo que las primeras llevan como primer parámetro un puntero a la ventana sobre la cual se desea trabajar.

Detallar cada una de las funciones abarcaría mucho espacio, por lo que recomiendo buscar información detallada sobre ellas en las páginas del manual que provee el sistema. Más adelante veremos un ejemplo.

Colores

Es posible utilizar colores en nuestros programas. Existe una función que debemos llamar para poder trabajar con lo colores, pero antes comprobar si existe soporte para los colores.

```
if(has_colors())
    start_color();
```

Ahora si, a utilizar los colores pero antes de eso debemos inicializar pares de colores, dicho par definirá el color del primer y segundo plano respectivamente. Esto lo hacemos con:

```
int init_pair(short pair, short f, short
b);
```

Esta función crea un par de colores y los identifica con un número, el primer argumento, este número servirá para hacer referencia a este par de colores cuando los necesitemos. Los 2 argumentos restantes son constantes definidas en la biblioteca **Ncurses** las cuales son del tipo **COLOR_BLACK, COLOR_BLUE**,etc. Para más información ver la documentación.

Después de haber definido los pares de colores a utilizar necesitamos llamar a otra función para indicar en que momento queremos activar cierto par de colores, después de hacer la llamada todo texto impreso en pantalla utilizará el par de colores activado.

```
attron(COLOR_PAIR(i));
```

Esta función activa el par de colores apuntado por i.

Para cambiar el color de fondo de una ventana utilizamos la función **bkgd()** para la ventana estándar y **wbkgd** para las otras ventanas.

```
int bkgd(chtype ch);
int wbkgd(WINDOW *win, chtype ch);
```

El argumento **chtype ch** es un par de colores previamente definidos, por ejemplo, **bkgd(COLOR_PAIR(1))** pone como fondo de la ventana estándar el color definido en el par número 1.

El cursor

Veamos el uso del cursor lógico, como mencionamos anteriormente, éste apunta a la posición, nosotros podemos manipular la posición del cursor para escribir en la posición de pantalla que deseamos La función **int move(int y, int x)** posiciona el cursor en la posición indicada por x e y, toda impresión que hagamos después de **move** iniciará en esta posición. **wmove** hace lo mismo pero para una ventana en especifico. Es curioso saber que **move(5,3)** es lo mismo que **wmove(stdscr, 5,3)**.

Menús

En realidad el efecto de los menús se emula mediante el uso de pequeñas subventanas que se crean y se eliminan cada vez que los queremos visualizar. En el ejemplo de más adelante lo veremos, por lo que lo dejamos pendiente.

Reconocimiento del teclado

Ncurses también tiene funciones para identificar que teclas presionamos en un determinado momento, es más, es posible utilizar las teclas de función en nuestros programas. Las siguientes funciones hacen esto posible.

La siguiente función permite la utilización de las teclas de función, primer parámetro, sobre que ventana, segundo parámetro, activar o desactivar esta función, 1 = activo, 0 = desactivado.

```
int keypad(WINDOW *win, bool bf);
```

Si tecleamos cuando esta activo nuestro programa, hay dos posibilidades, que en la pantalla aparezcan estos caracteres o que no se impriman, **echo** hace realidad la primera posibilidad y **noecho** la segunda.

```
int echo(void);
int noecho(void);
```

En el caso de la entrada de datos nuevamente hay 2 posibilidades, que los datos tecleados sean enviados al programa hasta después de presionar la tecla intro, enviar los datos inmediatamente después de teclearlos. Las 2 funciones siguientes activan y/o desactivan el envío inmediato de valores al programa. **cbreak** activa el envío inmediato.

```
int cbreak(void);
int nocbreak(void);
```

Para conocer la tecla que se ha presionado nos apoyamos del valor que devuelve la función **getch**, por poner un ejemplo. En la biblioteca **Ncurses** están definidas algunas macros para identificar algunas teclas especiales tales como las teclas de función con KEY_F(i), donde i es el número de la tecla de función, KEY_UP, para la tecla direccional que va hacia arriba, entre otras.

Finalización

Por último y como ya se había mencionado, para salir del modo gráfico tenemos que llamar a la función **endwin()** sin ningún parámetro para restablecer el modo consola.

A pesar de que no se ha mencionado, de seguro se preguntarán cual es el archivo de cabecera que debemos incluir para hacer uso de las funciones mencionadas, pues bien el archivo de cabecera es el **ncurses.h**, sin él no podremos utilizar nada de lo mencionado.

Ejemplo

Podríamos seguir describiendo funciones y más funciones pero no hay nada mejor que un ejemplo para observar la utilidad de lo que hemos visto, es un programa sencillo pero creo que es suficiente para empezar.

```
#include<stdlib.h>      //Para la reserva de memoria
#include<ncurses.h>     //Contiene todas las funciones de Ncurses
#define SALIR 27

void Inicializar();     //Para inicializar el modo grafico y algunos colores
void DibujaBarra(WINDOW *barra);//Para dibujar la barra de munus
WINDOW **DibujaMenu(int inicioCol);
void EliminaMenu(WINDOW **Elementos,int TamElems);
int DesplazaMenu(WINDOW **Elementos,int TamElems,int inicioMenuCol);

int main()
{
        int opt;
        WINDOW *barra;
        Inicializar();

        bkgd(COLOR_PAIR(1));
        barra = subwin(stdscr,1,80,0,0);
        DibujaBarra(barra);

        move(10,10);    //Nos posicionamos en la fila 10 y columna 10
        printw("Este es un programa que utilza la Biblioteca Ncurses. ");
        move(11,10);
        printw("Para activar el menu Presiona F5.");
        move(12,10);
        printw("Para salir del Menu presiona una ves la tecla ESCAPE.");
        move(13,10);
        printw("Para salir del programa presiona la tecla ESCAPE nuevamente.");

        refresh();      //Refrescamos la ventana, la estandar
        do{
                WINDOW **Elementos;
                opt=getch();
                if (opt==KEY_F(5)) {
                        Elementos = DibujaMenu(0);
                        DesplazaMenu(Elementos, 5, 0);
                        EliminaMenu(Elementos,6);
                        touchwin(stdscr);
            refresh();
        }
        }while(opt != SALIR);

        endwin();       //Finalizamos el modo grafico
        return 0;
}
//--------------------------------------------------------------------------------
void Inicializar()
{
        initscr();
    start_color();
    init_pair(1,COLOR_WHITE,COLOR_BLUE);//Creamos unos pares de colores
    init_pair(2,COLOR_BLUE,COLOR_WHITE);
    init_pair(3,COLOR_RED,COLOR_WHITE);
    curs_set(0);        //Para que no muestre el cursor fisico
    noecho();
    keypad(stdscr,1);//Activamos la recepcion de teclas de funcion
}
```

```
//----------------------------------------------------------------------------------
void DibujaBarra(WINDOW *barra)
{
        wbkgd(barra,COLOR_PAIR(2));
        waddstr(barra,"Archivo");//Esta funcion imprime una cadena de caracteres
        wattron(barra,COLOR_PAIR(3));//Cambiamos el color al texto que sigue
        waddstr(barra,"F5");
}
//----------------------------------------------------------------------------------
WINDOW **DibujaMenu(int inicioCol)
{
        int i;
    WINDOW **Elem;      //Un puntero a varios ventanas, un array
        //Reserva de memoria
    Elem = (WINDOW **)malloc(6*sizeof(WINDOW *));
    Elem[0]=newwin(10,19,1,inicioCol);
    wbkgd(Elem[0],COLOR_PAIR(2));
    box(Elem[0],ACS_VLINE,ACS_HLINE);   //Para dibujar un marco
    Elem[1]=subwin(Elem[0],1,17,2,inicioCol+1);
    Elem[2]=subwin(Elem[0],1,17,3,inicioCol+1);
    Elem[3]=subwin(Elem[0],1,17,4,inicioCol+1);
    Elem[4]=subwin(Elem[0],1,17,5,inicioCol+1);
    Elem[5]=subwin(Elem[0],1,17,6,inicioCol+1);


    for (i=1;i<6;i++)
        wprintw(Elem[i],"Opcion %d",i);
    wbkgd(Elem[1],COLOR_PAIR(1));
    wrefresh(Elem[0]);
    return Elem;
}//--------------------------------------------------------------------------------
int DesplazaMenu(WINDOW **Elementos,int TamElems,int inicioMenuCol)
{
        int tecla;
        int S = 0;
        while (1) {
            tecla=getch();
                //Si se presiono una tecla Arriba o Abajo
            if (tecla == KEY_DOWN || tecla == KEY_UP){
                wbkgd(Elementos[S+1],COLOR_PAIR(2));
                wnoutrefresh(Elementos[S+1]);
                if (tecla == KEY_DOWN) {    //Controlamos el desplazamiento
                    S = (S+1) % TamElems;   //de las teclas
                } else {
                    S = (S + TamElems - 1) % TamElems;
                }
                wbkgd(Elementos[S + 1],COLOR_PAIR(1));
                wnoutrefresh(Elementos[S + 1]);
                doupdate();//Sirve para refrescar la ventana
            }else if (tecla==SALIR){ return 0;  }
        }
}
//----------------------------------------------------------------------------------
void EliminaMenu(WINDOW **Elementos,int TamElems)
{
        int j;
        for(j = 0; j<TamElems; j++)
                delwin(Elementos[j]);
        free(Elementos);//Liberacion de memoria
}
```

Compilación

Para compilar los programas de éste tipo lo único que tenemos que hacer es que en el momento de compilación debemos indicar la biblioteca que debe llamar para crear el programa, esta biblioteca es **`libcurses`**, debido a la convención solo se escribe la parte después de **`lib`**. La línea de compilación quedaría de la siguiente forma.

```
gcc -Wall programafuente.c -o programaejecutable -lcurses
```

En nuestro caso si nombran al programa **`ventanas.c`** la línea de compilación quedaría así:

```
gcc -Wall ventanas.c -o ventanas -lcurses
```

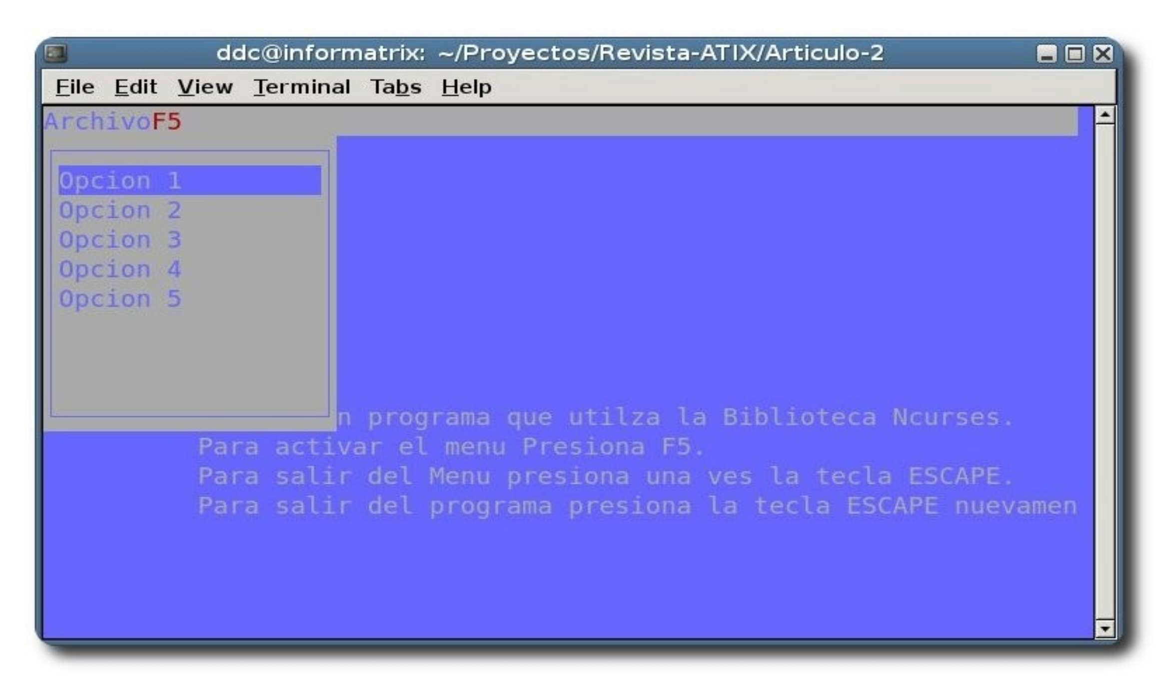


Gráfico 5: Resultado.

Documentación

Abusando un poco de la extensión del documento vamos a ver como podemos obtener información sobre estas funciones en las páginas de los manuales que proporciona la biblioteca **Ncurses**.

Las funciones de **Ncurses** se encuentran el la tercera sección de los manuales y para acceder a ellas deberíamos teclear el siguiente comando.

`man 3 funcion`

`Funcion` es el nombre de la función que deseamos buscar. En realidad ya en la práctica no siempre es necesario escribir el número 3, pero si existiesen colisiones con otras funciones del mismo nombre el número nos permite quitar esa ambigüedad.

```
ddc@informatrix: ~/Proyectos/Revista-ATIX/Articulo-2
File  Edit  View  Terminal  Tabs  Help
printw(3NCURSES)                                          printw(3NCURSES)

NAME
       printw, wprintw, mvprintw, mvwprintw, vwprintw, vw_printw -
       print formatted output in curses windows

SYNOPSIS
       #include <curses.h>

       int printw(const char *fmt, ...);
       int wprintw(WINDOW *win, const char *fmt, ...);
       int mvprintw(int y, int x, const char *fmt, ...);
       int mvwprintw(WINDOW *win, int y, int x,  const  char  *fmt,
       ...);
 Manual page printw(3ncurses) line 1
```

Gráfico 6: Documentación

Conclusión

Creo que con este pequeño y sencillo material es suficiente para que se den idea y puedan aprender más por su propia cuenta si el tema les interesa. Es cierto que faltaron muchas cosas que mencionar pero es parte de lo que ustedes pueden ahora aprender por si mismos.

Por cualquier duda, comentario o corrección sobre este artículo no duden en hacérmelo saber que con gusto atenderé.

Referencias

[1] http://www.gnu.org/software/ncurses/

[2] http://es.wikipedia.org/wiki/Ncurses

[3] http://es.wikipedia.org/wiki/Curses

Recursos importantes

[1] http://tldp.org/HOWTO/NCURSES-Programming-HOWTO/

[2] http://www.linuxfocus.org/Castellano/March2002/article233.shtml

Autor



Rafael Rendón Pablo
Estudiante de Ingeniería en Sistemas Computacionales
Usuario de software libre
smart.dyr@gmail.com



Desarrollo Ágil con Ruby on Rails (5ta Parte)

Ruby on Rails es un framework de desarrollo web ágil, elaborado por David Heinemeier Hansson, que el 2004 lanzó la versión pública a partir de desarrollar el proyecto Basecamp, Ruby on **Rails** (RoR) está desarrollado en el lenguaje de programación Ruby, RoR actualmente se encuentra en la versión 2.1.1 en la que contribuyeron más de 1400 desarrolladores del mundo con 1600 parches al framework, por estos datos RoR es uno de los proyectos open source con más movimiento actual.



Finalizando el ciclo de tutoriales sobre desarrollo ágil con el framework Ruby on **Rails**, añadiremos la característica de Ajax (Llamada asincrónica de javascript y xml), para mejorar los tiempos de respuesta de nuestra aplicación.

La funcionalidad que añadiremos la característica de Ajax, será en la ventana de login que tenemos, en resumen lo que se hará es que al momento de loguearse envié los datos mediante una llamada Ajax y nos muestre con una animación, un mensaje si los datos introducidos fueron correctos o no., Ahora empecemos:

Rails por defecto tiene helpers (o plantillas de código) para facilitarnos el aprendizaje y trabajo con ajax, además mediante su concepto de REST (todo es un recurso) que permite realizar una respuesta personalizada para este tipo de llamadas, para ésto maneja un tipo especial de fichero: los RJS que nos devuelven ficheros **.js**.

Para empezar incluiremos los ficheros necesarios para poder trabajar con Ajax, son las librerías prototipe y scriptaculous; **Rails** tiene una estupenda integración con estas librerías mediante helpers, bien para esto definimos los siguiente dentro de la plantilla de la página en **/app/views/layouts/application.html.erb**, añadimos la siguiente línea dentro de la etiqueta **<head>** de la página:

```
<%= javascript_include_tag :defaults %>
```

Con ésto creará las etiquetas necesarias y las llamadas a las librerías javascript para trabajar, ahora se modificará el tipo de formulario dentro de /app/views/sessions/new.html.erb para que realice llamadas ajax al momento de enviarlas, cambiamos el código por lo siguiente:

```
<h1>Iniciar sesión</h1>
<div id="mensaje" style="display:none;" >
   <%= image_tag 'spinner.gif'%> Cargando
</div>
<% form_remote_tag :url => session_path, :before => "$('mensaje').show()" do -%>
  <p><%= label_tag 'login' %>
  <%= text_field_tag 'login', @login %></p>
  <p><%= label_tag 'password' %>
  <%= password_field_tag 'password', nil %></p>
  <p><%= submit_tag 'Enviar' %></p>
<% end -%>
```

Ahora tenemos que generar la respuesta dentro del controlador, para este tipo de llamada, para eso nos vamos a **/app/controllers/sessions_controller.rb** y modificar todo el método **create**, de la siguiente manera:

```
  def create
    logout_keeping_session!
    @user = User.authenticate(params[:login], params[:password])
    if @user
      self.current_user = @user
      new_cookie_flag = (params[:remember_me] == "1")
      handle_remember_cookie! new_cookie_flag
      flash[:notice] = "Logged in successfully"
      respond_to do |format|
        format.html { redirect_back_or_default('/') }
        format.js
      end
    else
      note_failed_signin
      @login       = params[:login]
      @remember_me = params[:remember_me]
      respond_to do |format|
        format.html { render :action => 'new' }
        format.js
      end
    end
```

Con ésto le decimos a esta acción que responda a dos tipos de recursos que pidamos, el normal o **html** y el nuevo que es ajax o **js**.

Bien ahora tenemos que elaborar la respuesta js, **Rails** maneja esta respuesta mediante un fichero del tipo **{accion}.rjs**, en este caso un fichero es **/app/views/sessions/create.rjs** que tendrá lo siguiente:

```
if @user
  page[:mensaje].replace_html('Se logueo satisfactoriamente.')
  page << "document.location.href = '/revistas';"
else
  page[:mensaje].replace_html('Usuario y contraseña equivocados.')
end
page[:mensaje].visual_effect :Highlight, :startcolor => '#FFFF99'
flash.discard
```

Este es una muestra del funcionamiento:



La respuesta que genera es javascript puro en el que: Si no se logueo satisfactoriamente, muestre un mensaje de “Usuario y contraseña equivocados” y si ingreso los datos correctos, que muestre el mensaje “Se logueo satisfactoriamente” y que redireccione mediante javascript al listado de revistas.

Quiero recalcar que el código es muy trivial y es para propósitos de estudio, la funcionalidad de ajax se la puede añadir a cualquier tipo de acción en común, pero no abusar ya que causaría confusión al usuario, porque en común la respuesta es inmediata, por eso es importante mostrar algún mensaje o efecto, como en el ejemplo, el código de este artículo se encuentra en **`http://github.com/carakan/atix`** del cual pueden bajarlo y analizarlo.

Bien siendo mi último artículo dentro de la revista ATIX, del cual estoy muy honrado de colaborar, espero que todo lo abordado sea de ayuda para realizar los primeros pasos dentro de este mundo railero, ahora quiero recomendarles sitios web donde podrán aprender más acerca de **Rails**, uno de ellos y es el que por defecto tenemos que ver es **`http://railscasts.com/`**, del que semanalmente se muestran diferentes tips y consejos para el desarrollo diario con **Rails**, asimismo formamos una comunidad en torno al desarrollo con **Rails**, que se encuentra en **`http://boliviaonrails.com`** del cual están invitados a formar parte y/o colaborar.

Referencias

[1] http://www.rubyon**rails**.org/

[2] http://www.rubyforge.org/

Autor



Carlos Ramos

Lic. Informática UMSA
Lider de Wiebia, soluciones web 2.0
carakan@gmail.com
Blog personal http://ww.carakan.com
http://www.wiebia.com

Grails: Framework para el desarrollo de aplicaciones Web (2da Parte)

Grails es un framework orientado al desarrollo de aplicaciones web de forma sencilla, rápida y divertida.



En la primera parte del artículo, se vio la creación de la aplicación y la generación de scaffoldings dinámicos, en ésta segunda entrega veremos la generación del scaffolding estático, las validaciones en los formularios, personalizaciones de mensajes y las personalizaciones de vistas.

Scaffolding estático

La diferencia entre un scaffolding dinámico y un scaffolding estático, radica que el primero genera el código necesario de la aplicación en tiempo de ejecución, mientras que el segundo genera el código antes de la ejecución de la aplicación, permitiéndonos de esta forma realizar personalizaciones en el código de nuestra aplicación.

Generación de scaffolding estático

La generación de scaffold puede incluir todos los componentes (vistas y controladores) o solo los componentes necesarios ya sean vistas o controladores.

Para la generación de los scaffold estáticos debemos hacer uso de la instrucción **`grails generate-all`**, tal como muestra la figura:

```
jesaavedra@desarrollo: ~/GestionEventos
Archivo  Editar  Ver  Terminal  Solapas  Ayuda
jesaavedra@desarrollo:~/GestionEventos$
jesaavedra@desarrollo:~/GestionEventos$ grails generate-all Evento

Welcome to Grails 1.0.4 - http://grails.org/
Licensed under Apache Standard License 2.0
Grails home is set to: /home/jesaavedra/aplic_web/grails

Base Directory: /home/jesaavedra/GestionEventos
Running script /home/jesaavedra/aplic_web/grails/scripts/GenerateAll.groovy
Environment set to development
Loading with installed plug-ins: ["acegi", "jsecurity"] ...
     [copy] Copying 1 file to /home/jesaavedra/.grails/1.0.4/projects/GestionEventos
[0] spring.GrailsWebApplicationContext Refreshing org.codehaus.groovy.grails.commons.spring.Grai
lsWebApplicationContext@1c2d5534: display name [org.codehaus.groovy.grails.commons.spring.Grails
WebApplicationContext@1c2d5534]; startup date [Thu Mar 26 10:06:41 BOT 2009]; root of context hi
erarchy
[0] spring.GrailsWebApplicationContext Bean factory for application context [org.codehaus.groovy
.grails.commons.spring.GrailsWebApplicationContext@1c2d5534]: org.springframework.beans.factory.
support.DefaultListableBeanFactory@7b4653a3
     [copy] Copying 1 file to /home/jesaavedra/.grails/1.0.4/projects/GestionEventos
Loading with installed plug-ins: ["acegi", "jsecurity"] ...
Generating views for domain class Evento ...
Generating controller for domain class Evento ...
Finished generation for domain class Evento
jesaavedra@desarrollo:~/GestionEventos$
jesaavedra@desarrollo:~/GestionEventos$
```

Gráfico 1: Generación de scaffolding estático

Esta generación se debe realizar de forma análoga para los demás dominios, en nuestro caso para Expositor y para Conferencia.

Dentro del directorio **GestionEventos/grails-app/controllers** y **GestionEventos/grails-app/views**, se encuentra el código generado para los controladores y las vistas respectivamente.

```
jesaavedra@desarrollo: ~/GestionEventos/grails-app
Archivo  Editar  Ver  Terminal  Solapas  Ayuda
jesaavedra@desarrollo:~/GestionEventos/grails-app$
jesaavedra@desarrollo:~/GestionEventos/grails-app$ tree controllers/
controllers/
|-- ConferenciaController.groovy
|-- EventoController.groovy
`-- ExpositorController.groovy

0 directories, 3 files
jesaavedra@desarrollo:~/GestionEventos/grails-app$
```

Gráfico 2: Controladores generados

```
jesaavedra@desarrollo: ~/GestionEventos/grails-app
Archivo  Editar  Ver  Terminal  Solapas  Ayuda
jesaavedra@desarrollo:~/GestionEventos/grails-app$
jesaavedra@desarrollo:~/GestionEventos/grails-app$ tree views/
views/
|-- conferencia
|   |-- create.gsp
|   |-- edit.gsp
|   |-- list.gsp
|   `-- show.gsp
|-- error.gsp
|-- evento
|   |-- create.gsp
|   |-- edit.gsp
|   |-- list.gsp
|   `-- show.gsp
|-- expositor
|   |-- create.gsp
|   |-- edit.gsp
|   |-- list.gsp
|   `-- show.gsp
`-- layouts
    `-- main.gsp

4 directories, 14 files
jesaavedra@desarrollo:~/GestionEventos/grails-app$
```

Gráfico 3: Vistas generadas

Conexión a una base de datos

Todos deseamos que nuestras aplicaciones web se conecten en algún momento a una base de datos (MySQL, PostgreSQL, etc), para ésto precisamos realizar lo siguiente:

- ✔ Incluir el controlador JDBC correspondiente a la base de datos en el directorio **GestionEventos/lib**; para nuestro caso haremos uso del controlador **mysql-connector-java-5.1.5-bin.jar**, que nos permitirá establecer la conexión a una base de datos mySQL.
- ✔ Modificar las cadenas de conexión (base de datos, usuario, driver, etc) en el fichero **GestionEventos/grails-app/conf/DataSource.groovy**.

Un ejemplo de este archivo es el siguiente:

```
dataSource {
        pooled = true
        driverClassName = "com.mysql.jdbc.Driver"
        username = "usuario"
        password = "clave"
}
hibernate {
    cache.use_second_level_cache=true
    cache.use_query_cache=true
    cache.provider_class='com.opensymphony.oscache.hibernate.OSCacheProvider'
}
// environment specific settings
environments {
        development {
                dataSource {
                        dbCreate = "update" // one of 'create', 'create-drop','update'
                        url = "jdbc:mysql://localhost:3306/eventos"
                }
        }
        test {
                dataSource {
                        dbCreate = "update"
                        url = "jdbc:mysql://localhost:3306/eventos"
                }
        }
        production {
                dataSource {
                        dbCreate = "update"
                        url = "jdbc:mysql://localhost:3306/eventos"
                }
        }
}
```

Una vez realizada las adecuación del archivo que permite la conexión a la base de datos, debemos proceder a crear la base de datos y sus respectivas tablas, para esto podemos hacer uso de cualquier manejador de mySQL.

Validación de formularios

Una de las tareas importantes al momento de realizar un formulario web, es la validación de los campos; dentro del framework Grails, ésta es conocida como **constrains**, y deben ser consideradas dentro de los archivos de cada dominio. Un ejemplo de **constrains** es el siguiente:

```
class Expositor {
   String paterno
   String materno
   String nombre
   String nacionalidad
   String email

   static hasMany = [conferencias:Conferencia]

   static constraints={
   paterno (blank:false,size:1..25)
   materno (blank:false,size:1..25)
   nombre (blank:false,size:1..25)
```

```
    email (email:true)
    nacionalidad (inList:['Argentina','Bolivia','Brasil','Chile','Colombia'])
}
}
```

Valores por defecto

Si observamos en los formularios que contienen relaciones, éstas son representadas por medio de persianas desplegables, donde los valores aparecen como **`<nombre de la tabla>:<ID>`**; como se muestra en la figura:

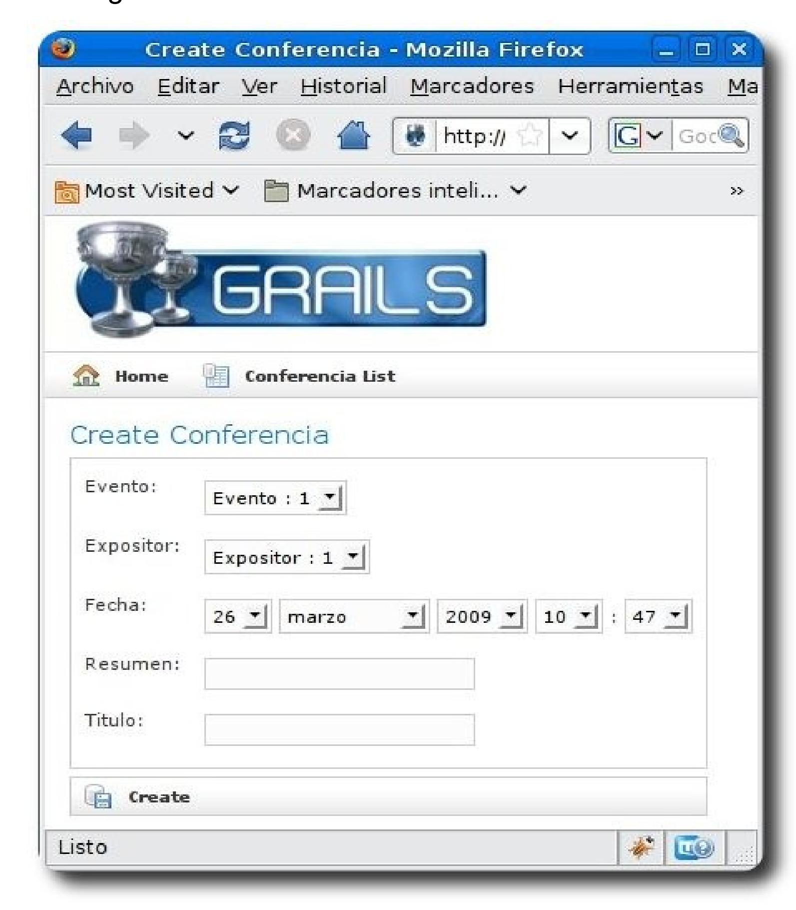


Gráfico 4: Valores por defecto en persianas desplegables

Para que ésto cambie de apariencia y muestre el valor de uno de los campos de la tabla o la combinación de campos, debemos crear un método que devuelva un determinado valor cada vez que se haga referencia a un objeto, por ejemplo si realizamos los siguientes cambios:

`GestionEventos/grails-app/domain/Expositor.groovy`

```
class Expositor {
    String paterno
    String materno
    String nombre
    String nacionalidad
    String email
    static hasMany = [conferencias:Conferencia]
    String toString ()
    {
     return nombre + " " + paterno
    }
}
```

GestionEventos/grails-app/domain/Evento.groovy

```
class Evento {
   String nombre
   String ciudad
   Date inicia
   Date finaliza
   String sitioweb
   static hasMany = [conferencias:Conferencia]
   String toString ()
    {
     return nombre
    }
}
```

En este ejemplo hará que las persianas desplegables muestren el nombre del Expositor y del Evento, en todos los formularios con los que mantenga relación, tal como muestra la figura siguiente:

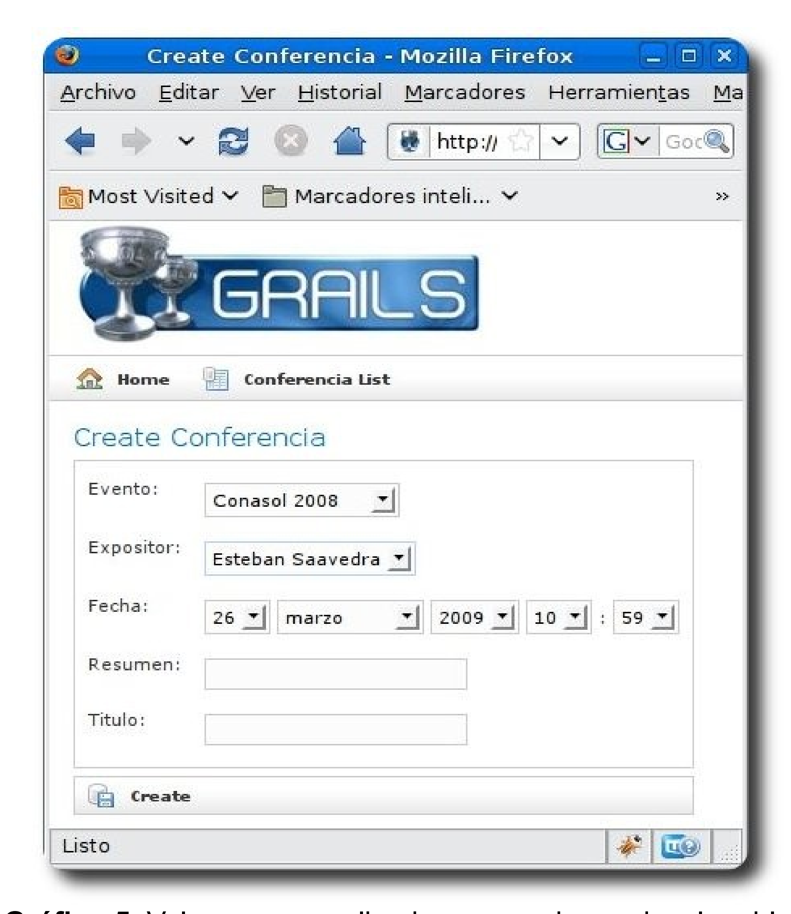


Gráfico 5: Valores personalizados en persianas desplegables

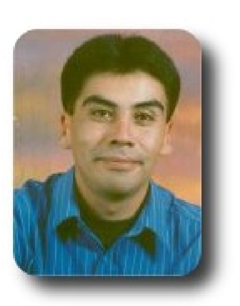
Referencias

[1] http://www.grails.org

[2] Getting Started with Grails

[3] The Definitive Guide to Grails

Autor



Esteban Saavedra López
Líder de la Comunidad ATIX (Oruro – Bolivia)
Activista de Software Libre en Bolivia
jesaavedra@opentelematics.org
http://jesaavedra.opentelematics.org

Monitoreo de servidores mediante SNMP

Cuando tenemos varios equipos en nuestra red, es necesario poder monitorearlos, la forma más básica de monitoreo por red es mediante el Protocolo SNMP (Simple Network Management Protocol o Protocolo simple de administración de red), en este artículo veremos la forma de configurar la herramienta NET-SNMP en un equipo LINUX para poder monitorearlo por la red.



Introducción

El Protocolo Simple de Administración de Red o SNMP es un protocolo que permite la administración de equipos de una red, desde el simple monitoreo, pasando por el cambio de configuraciones hasta el encendido y apagado de estos equipos, los mismos pueden ser: Servidores, equipos de comunicaciones, impresoras de red, UPS, etc.

En este artículo veremos la forma básica de configurar la herramienta NET-SNMP en un equipo con LINUX para poder monitorearlo por la red. La herramienta NET-SNMP es una suite de programas que permiten el manejo de SNMP en los equipos bajo LINUX.

Instalación de NET-SNMP

Utilizaremos la distribución DEBIAN para implementar el NET-SNMP.

Para la instalación ejecutaremos el siguiente comando:

```
# apt-get install snmp snmpd scli tkmib
```

Donde los paquetes que utilizaremos son:

- ✔ **snmpd**: Demonio SNMP.
- ✔ **snmp**: Comandos para clientes SNMP.
- ✔ **scli**: Comandos para monitoreo y configuración SNMP.
- ✔ **tkmib**: Interfaz gráfica.

Configuración SNMP

Los archivos de configuración están en el directorio **`/etc/snmp`**.

Los archivos de configuración son **`snmpd.conf`** y **`snmptrapd.conf`**.

Donde:

- ✔ **`/etc/snmp/snmpd.conf`** : Archivo de configuración del NET-SNMP agente.
- ✔ **`/etc/snmp/snmptrapd.conf`** : Archivo de configuración del Net-SNMP demonio TRAP

La configuración básica la realizaremos sobre el archivo **`snmpd.conf`**.

Para configurar el servidor SNMP y habilitar el acceso de lectura desde otras máquinas por la red, debemos modificar el archivo **`/etc/snmp/snmpd.conf`** de la siguiente manera:

Habilitar las conexiones a la comunidad SNMP

Cambiamos de:

```
####
# First, map the community name (COMMUNITY) into a security name
# (local and mynetwork, depending on where the request is coming
# from):

#         sec.name  source          community
com2sec paranoid  default         public
#com2sec readonly  default         public
#com2sec readwrite default         private
```

a:

```
####
# First, map the community name (COMMUNITY) into a security name
# (local and mynetwork, depending on where the request is coming
# from):

#         sec.name  source          community
#com2sec  paranoid  default          public
com2sec  local     localhost       test01
com2sec  localnet  172.16.16.0/24  test01
#com2sec readwrite default         private
```

Con este cambio daremos permisos para el acceso local y desde el segmento de red **`172.16.16.0/24`** al servidor por medio del protocolo SNMP.

El nombre de la comunidad SNMP por defecto es "**`public`**" y en este ejemplo se lo cambio a "**`test01`**", cuando se habilita el servicio de SNMP se recomienda cambiar el nombre de la comunicad, por razones de seguridad.

Habilitar los nombres de seguridad a los nombres de grupo

Cambiamos de:

```
####
# Second, map the security names into group names:
#                sec.model  sec.name
group MyROSystem v1        paranoid
group MyROSystem v2c       paranoid
group MyROSystem usm       paranoid
group MyROGroup v1         readonly
group MyROGroup v2c        readonly
group MyROGroup usm        readonly
group MyRWGroup v1         readwrite
group MyRWGroup v2c        readwrite
group MyRWGroup usm        readwrite
```

a:

```
####
# Second, map the security names into group names:
#               sec.model  sec.name
group MyROSystem v1        local
group MyROSystem v2c       local
group MyROSystem usm       local
group MyROGroup v1         localnet
group MyROGroup v2c        localnet
group MyROGroup usm        localnet
group MyRWGroup v1         local
group MyRWGroup v2c        local
group MyRWGroup usm        local
```

Con este cambio damos permisos de lectura y escritura a las conexiones locales y damos permisos solo de lectura a conexiones desde el segmento de red **172.16.16.0/24**. Es decir que desde una conexión local se podrán realizar consultas SNMP y realizar cambios a la configuración del servidor, y desde una conexión del segmento de red **172.16.16.0/24** solo se podrán realizar consultas SNMP.

Habilitar el monitoreo de file systems y de rendimiento del equipo

Cambiamos de:

```
####
# Third, create a view for us to let the groups have rights to:
#           incl/excl subtree                          mask
view all    included  .1                               80
view system included  .iso.org.dod.internet.mgmt.mib-2.system
```

a:

```
####
# Third, create a view for us to let the groups have rights to:
#           incl/excl subtree                          mask
view all    included  .1                               80
view system included  .iso.org.dod.internet.mgmt.mib-2.system
view system included .1.3.6.1.4.1.2021
```

Con este cambio habilitamos el poder ver los MIBS de monitoreo de FILE SYSTEM y de RENDIMIENTO.

Como ejemplo, para habilitar el monitoreo del File System / (root) realizamos el siguiente cambio de:

```
###################################################################
# disk checks
#

# The agent can check the amount of available disk space, and make
# sure it is above a set limit.

# disk PATH [MIN=DEFDISKMINIMUMSPACE]
#
# PATH:  mount path to the disk in question.
# MIN:   Disks with space below this value will have the Mib's errorFlag set.
#        Default value = DEFDISKMINIMUMSPACE.

# Check the / partition and make sure it contains at least 10 megs.

disk / 10000
```

a:

```
###################################################################
# disk checks
#

# The agent can check the amount of available disk space, and make
# sure it is above a set limit.

# disk PATH [MIN=DEFDISKMINIMUMSPACE]
#
# PATH:  mount path to the disk in question.
# MIN:   Disks with space below this value will have the Mib's errorFlag set.
#        Default value = DEFDISKMINIMUMSPACE.

# Check the / partition and make sure it contains at least 10 megs.

disk / 512000
```

Donde:

512000 es el límite de espacio libre que puede tener el File System /, si el valor de espacio libre esta por debajo de este valor se generará una alarma por SNMP, este valor está expresado en kilo bytes, en el ejemplo 500MB.

Para poder habilitar el monitoreo del rendimiento del CPU se debe cambiar de:

```
###################################################################
# load average checks
#

# load [1MAX=DEFMAXLOADAVE] [5MAX=DEFMAXLOADAVE] [15MAX=DEFMAXLOADAVE]
#
# 1MAX:   If the 1 minute load average is above this limit at query
#         time, the errorFlag will be set.
# 5MAX:   Similar, but for 5 min average.
# 15MAX:  Similar, but for 15 min average.

# Check for loads:
#load 12 14 14
```

a:

```
####################################################################
# load average checks
#

# load [1MAX=DEFMAXLOADAVE] [5MAX=DEFMAXLOADAVE] [15MAX=DEFMAXLOADAVE]
#
# 1MAX:   If the 1 minute load average is above this limit at query
#         time, the errorFlag will be set.
# 5MAX:   Similar, but for 5 min average.
# 15MAX:  Similar, but for 15 min average.

# Check for loads:
load 10 15 15
```

Donde:

- ✔ 10 es el límite de porcentaje de carga promedio en un minuto, si el valor de porcentaje de carga promedio está por debajo de este valor se generará una alarma por SNMP.
- ✔ 15 es el límite de porcentaje de carga promedio en cinco minutos, si el valor de porcentaje de carga promedio está por debajo de este valor se generará una alarma por SNMP.
- ✔ 15 es el límite de porcentaje de carga promedio en quince minutos, si el valor de porcentaje de carga promedio está por debajo de este valor se generará una alarma por SNMP.

Reincio del servicio snmpd

Para reiniciar el servicio de snmpd se debe ejecutar el siguiente comando:

```
# /etc/init.d/snmpd restart
Restarting network management services: snmpd.
#
```

Consultas snmp (monitoreo)

Una vez terminada la configuración del Servicio SNMP podremos realizar las consultas (lecturas) de la información enviada por este protocolo. Con estas consultas podremos monitorear la salud del equipo.

Las consultas (lecturas) SNMP se las puede realizar de dos formas:

- ✔ Por información de un solo objeto, a través del comando snmpget.
- ✔ Por árbol de información (varios objetos), a través del comando snmpwalk.

Estos comandos se los puede ejecutar desde el mismo equipo, desde otro equipo LINUX que tenga instalado el paquete snmp (Comandos para clientes SNMP) o desde cualquier otro equipo que pueda realizar consultas SNMP (equipos de Monitoreo por red, equipos de comunicaciones para monitoreo SNMP, etc.).

A continuación realizamos dos consultas simples, una sobre la información del servidor Nombre del Sistema y la otra sobre la Descripción del Sistema, para ésto ejecutamos los siguientes comandos:

```
# snmpget -v1 -c test01 localhost .1.3.6.1.2.1.1.5.0
SNMPv2-MIB::sysName.0 = STRING: SRVDEBIANLAB1
#

# snmpget -v1 -c test01 localhost .1.3.6.1.2.1.1.1.0
SNMPv2-MIB::sysDescr.0 = STRING: Linux SRVDEBIANLAB1 2.4.27-2-686 #1 Mon May 16
17:03:22 JST 2005 i686
#
```

Ahora, realizaremos la consulta sobre todo el árbol de información del File System / (root):

```
# snmpwalk -v1 -c test01 localhost .1.3.6.1.4.1.2021.9
UCD-SNMP-MIB::dskIndex.1 = INTEGER: 1
UCD-SNMP-MIB::dskPath.1 = STRING: /
UCD-SNMP-MIB::dskDevice.1 = STRING: /dev/mapper/rootvg-rootlv
UCD-SNMP-MIB::dskMinimum.1 = INTEGER: 512000
UCD-SNMP-MIB::dskMinPercent.1 = INTEGER: -1
UCD-SNMP-MIB::dskTotal.1 = INTEGER: 5160576
UCD-SNMP-MIB::dskAvail.1 = INTEGER: 4216404
UCD-SNMP-MIB::dskUsed.1 = INTEGER: 682028
UCD-SNMP-MIB::dskPercent.1 = INTEGER: 14
UCD-SNMP-MIB::dskPercentNode.1 = INTEGER: 3
UCD-SNMP-MIB::dskErrorFlag.1 = INTEGER: 0
UCD-SNMP-MIB::dskErrorMsg.1 = STRING:
#
```

Donde podemos observar que en este ejemplo, el espacio total del File System / (root) es de 5160576KB (5GB aprox.) y el espacio ocupado es de 243684KB (238MB aprox.) el cual representa el 5% de utilización del total de espacio del File System.

Finalmente, realizaremos la consulta sobre todo el árbol de información del rendimiento del CPU:

```
# snmpwalk -v1 -c test01 localhost .1.3.6.1.4.1.2021.10
UCD-SNMP-MIB::laIndex.1 = INTEGER: 1
UCD-SNMP-MIB::laIndex.2 = INTEGER: 2
UCD-SNMP-MIB::laIndex.3 = INTEGER: 3
UCD-SNMP-MIB::laNames.1 = STRING: Load-1
UCD-SNMP-MIB::laNames.2 = STRING: Load-5
UCD-SNMP-MIB::laNames.3 = STRING: Load-15
UCD-SNMP-MIB::laLoad.1 = STRING: 1.15
UCD-SNMP-MIB::laLoad.2 = STRING: 1.03
UCD-SNMP-MIB::laLoad.3 = STRING: 0.71
UCD-SNMP-MIB::laConfig.1 = STRING: 10.00
UCD-SNMP-MIB::laConfig.2 = STRING: 15.00
UCD-SNMP-MIB::laConfig.3 = STRING: 15.00
UCD-SNMP-MIB::laLoadInt.1 = INTEGER: 114
UCD-SNMP-MIB::laLoadInt.2 = INTEGER: 103
UCD-SNMP-MIB::laLoadInt.3 = INTEGER: 70
UCD-SNMP-MIB::laLoadFloat.1 = Opaque: Float: 1.150000
UCD-SNMP-MIB::laLoadFloat.2 = Opaque: Float: 1.030000
UCD-SNMP-MIB::laLoadFloat.3 = Opaque: Float: 0.710000
UCD-SNMP-MIB::laErrorFlag.1 = INTEGER: 0
UCD-SNMP-MIB::laErrorFlag.2 = INTEGER: 0
UCD-SNMP-MIB::laErrorFlag.3 = INTEGER: 0
UCD-SNMP-MIB::laErrMessage.1 = STRING:
UCD-SNMP-MIB::laErrMessage.2 = STRING:
UCD-SNMP-MIB::laErrMessage.3 = STRING:
#
```

Otros ejemplos de consultas pueden ser el monitoreo de la memoria RAM y del área de paginamiento (SWAP):

```
# snmpwalk -v1 -c test01 localhost .1.3.6.1.4.1.2021.4
UCD-SNMP-MIB::memIndex.0 = INTEGER: 0
UCD-SNMP-MIB::memErrorName.0 = STRING: swap
UCD-SNMP-MIB::memTotalSwap.0 = INTEGER: 524280
UCD-SNMP-MIB::memAvailSwap.0 = INTEGER: 502496
UCD-SNMP-MIB::memTotalReal.0 = INTEGER: 256652
UCD-SNMP-MIB::memAvailReal.0 = INTEGER: 13208
UCD-SNMP-MIB::memTotalFree.0 = INTEGER: 515704
UCD-SNMP-MIB::memMinimumSwap.0 = INTEGER: 16000
UCD-SNMP-MIB::memShared.0 = INTEGER: 0
UCD-SNMP-MIB::memBuffer.0 = INTEGER: 107440
UCD-SNMP-MIB::memCached.0 = INTEGER: 32304
UCD-SNMP-MIB::memSwapError.0 = INTEGER: 0
UCD-SNMP-MIB::memSwapErrorMsg.0 = STRING:
#
```

y por último realizaremos la consulta sobre las Estadísticas del Sistema:

```
# snmpwalk -v1 -c test01 localhost .1.3.6.1.4.1.2021.11
UCD-SNMP-MIB::ssIndex.0 = INTEGER: 1
UCD-SNMP-MIB::ssErrorName.0 = STRING: systemStats
UCD-SNMP-MIB::ssSwapIn.0 = INTEGER: 0
UCD-SNMP-MIB::ssSwapOut.0 = INTEGER: 0
UCD-SNMP-MIB::ssIOSent.0 = INTEGER: 1
UCD-SNMP-MIB::ssIOReceive.0 = INTEGER: 7
UCD-SNMP-MIB::ssSysInterrupts.0 = INTEGER: 3
UCD-SNMP-MIB::ssSysContext.0 = INTEGER: 21
UCD-SNMP-MIB::ssCpuUser.0 = INTEGER: 0
UCD-SNMP-MIB::ssCpuSystem.0 = INTEGER: 0
UCD-SNMP-MIB::ssCpuIdle.0 = INTEGER: 98
UCD-SNMP-MIB::ssCpuRawUser.0 = Counter32: 857488
UCD-SNMP-MIB::ssCpuRawNice.0 = Counter32: 18565
UCD-SNMP-MIB::ssCpuRawSystem.0 = Counter32: 1391250
UCD-SNMP-MIB::ssCpuRawIdle.0 = Counter32: 193423693
UCD-SNMP-MIB::ssCpuRawKernel.0 = Counter32: 1391250
UCD-SNMP-MIB::ssIORawSent.0 = Counter32: 2232652
UCD-SNMP-MIB::ssIORawReceived.0 = Counter32: 26782300
UCD-SNMP-MIB::ssRawInterrupts.0 = Counter32: 220636758
UCD-SNMP-MIB::ssRawContexts.0 = Counter32: 83529399
UCD-SNMP-MIB::ssRawSwapIn.0 = Counter32: 6012
UCD-SNMP-MIB::ssRawSwapOut.0 = Counter32: 11400
#
```

Configuraciones adicionales

Adicionalmente se puede configurar el monitoreo a nivel de procesos, es decir que si quisieras monitorear algún proceso ó procesos específicos del sistema operativo, bases de datos ó aplicaciones podrás realizarlo configurando en:

```
#################################################################
# Process checks.
#
#  The following are examples of how to use the agent to check for
#  processes running on the host.  The syntax looks something like:
#
#  proc NAME [MAX=0] [MIN=0]
#
#  NAME:  the name of the process to check for.  It must match
#         exactly (ie, http will not find httpd processes).
#  MAX:   the maximum number allowed to be running.  Defaults to 0.
#  MIN:   the minimum number to be running.  Defaults to 0.

#
#  Examples:
#
#  Make sure mountd is running
#proc mountd
#  Make sure there are no more than 4 ntalkds running, but 0 is ok too.
#proc ntalkd 4
#  Make sure at least one sendmail, but less than or equal to 10 are running.
#proc sendmail 10 1
```

También se puede monitorear por medio de scripts, pudiendo de esta forma personalizar más el monitoreo. Estos scripts se los puede configurar en:

```
#################################################################
# Executables/scripts
#

#
#  You can also have programs run by the agent that return a single
#  line of output and an exit code.  Here are two examples.
#
#  exec NAME PROGRAM [ARGS ...]
#
#  NAME:     A generic name.
#  PROGRAM:  The program to run.  Include the path!
#  ARGS:     optional arguments to be passed to the program

# a simple hello world
#exec echotest /bin/echo hello world

# Run a shell script containing:
#
# #!/bin/sh
# echo hello world
# echo hi there
# exit 35
#
# Note:  this has been specifically commented out to prevent
# accidental security holes due to someone else on your system writing
# a /tmp/shtest before you do.  Uncomment to use it.
#
#exec shelltest /bin/sh /tmp/shtest
```

Conclusiones

Como pudimos observar en las consultas SNMP, es posible monitorear varios aspectos del equipo, principalmente los cuatro puntos que determinan la salud del equipo los cuales son: CPU, Memoria, Discos (File Systems) y Tráfico de red.

También se vio que la posibilidad de monitorear un equipo a través de SNMP es fácil y completa.

Si bien en este artículo solo se hablo del monitoreo a través de línea de comando; lo cual es tedioso al momento del monitoreo en línea (online); existen varias herramientas que pueden realizar el monitoreo en línea automáticamente y pueden guardar un histórico de la salud del equipo, también existen herramientas que cuentan con una consola de monitoreo (DASHBOARD) la cual nos permite ver la salud del equipo de un solo vistazo, estas herramientas pueden enviar alertas por SMTP (correo electrónico) ó mensajes a celulares (SMS).

Referencias

[1] http://www.debian.org/

[2] http://www.net-snmp.org

Autor



Renzo Martinez Pardo
Ingeniero Sistemas Electrónicos
renzomp@lycos.com

Rsync una excelente forma de respaldar y sincronizar tu información

Rsync es una aplicación para sistemas de tipo Unix que ofrece transmisión eficiente de datos incrementales comprimidos y cifrados. Mediante una técnica de delta encoding, permite sincronizar archivos y directorios entre dos máquinas de una red o entre dos ubicaciones en una misma máquina, minimizando el volúmen de datos transferidos.



Introducción

Actualmente una de las tareas más importantes de un administrador de sistemas, es la de realizar copias de seguridad, o mantener varios equipos con un mismo contenido (sincronización de servidores). Hoy en día hay multitud de aplicaciones comerciales y libres que nos permiten realizar éstas con más o menos capacidades.

Rsync

Es una aplicación que viene en todas las distribuciones de GNU/Linux. Su principal utilidad es la de sincronizar estructuras de árboles de directorios a través de la red, obviamente también puede ser utilizada dentro de un solo equipo.

Las amplias características que presenta rsync, permiten realizar multitud de acciones como:

- ✔ Copias locales
- ✔ Copias de servidores remotos
- ✔ Sistemas de mirroring
- ✔ Mantenimiento sincronizado de sistemas de pre-producción y producción, etc…

Características de rsync

- ✔ Facilidad en su uso y configuración
- ✔ Uso de un algoritmo que solo envía la información que ha cambiado en cada archivo, en lugar de enviarlo de forma completa.
- ✔ Permite comprimir la información enviada, facilitando el uso adecuado del ancho de banda
- ✔ Permite habilitar módulos (repositorios) con características comunes (a todos) y específicas (en particular)
- ✔ Permite la asignación de controles de acceso a los repositorios habilitados.
- ✔ Permite listar archivos de repositorios remotos
- ✔ Hace uso del protocolo de autenticación basado en un MD4 de 128 bits, permitiendo a su vez trabajar sobre ssh.

Instalación

La instalación (en caso que no estuviera instalado por defecto), se la hace mediante el gestor de paquetes propio de cada distribución:

- **RHEL, CentOS, Fedora**: yum install rsync
- **Debian, Ubuntu**: apt-get install rsync

Configuración de servidor y de clientes

Antes de proceder a la configuración tanto del servidor como de los clientes, es preciso detenernos a pensar cuál será la estructura de sincronización que existirá entre estos, cuáles serán los repositorios, si éstos serán públicos y solo accesibles para algunos usuarios, etc. estos detalles trataremos de explicarlos según se vayan dando cada caso dentro de la configuración del servidor.

Configurar Servidor

La mayoría de las opciones a configurar se encuentran en el lado del servidor, para esto se deben seguir los siguientes pasos:

- Crear un archivo de configuración (**`/etc/rsyncd.conf`**)
- Habilitar las opciones que permitirán la transferencia de archivos:
- Crear el archivo de usuarios y contraseñas, utilizado para la autenticación.
- Crear configuración del cliente

`/etc/rsyncd.conf`

```
# Ejecuto transferencias como usuario no root
iuid = rocio
gid = rocio
log file = /var/log/rsyncd.log
pid file = /var/run/rsyncd.pid

# Haciendo un chroot al directorio de los datos transferidos
use chroot = yes

# Varios
max connections = 4
syslog facility = daemon

# Permito acceso solo a la red local
hosts allow = 192.168.1.0/24
hosts deny = *

# Especificación de repositorios a publicar
[presentacion]
        path = /home/rocio/dir_presentacion
        comment = directorio publico de presentaciones

[proyecto]
        path = /home/rocio/dir_proyecto
        comment = directorio privado de proyectos
        read only = yes
        #No se permite listar
        #list = false
        #Solo permitido desde esta ip
        #hosts allow = 192.168.1.2
        auth users = rocio, anibal
        secrets file = /etc/rsyncd.secrets
        uid = rocio
        gid = rocio
```

Archivo de usuarios/contraseñas

Crear un archivo que contenga la lista de usuarios permitidos para acceder al servicio (**/etc/rsyncd.secrets**), este archivo es necesario solo en caso de hacer uso de las opciones: **auth users** y **secrets file**.

/etc/rsyncd.secrets

```
rocio:contraseña_de_rocio
anibal:contraseña_de_anibal
```

Explicación de la configuración

El archivo **/etc/rsyncd.conf** en su primera parte se compone de la configuración de parámetros genéricos del servidor, y una segunda parte donde se muestra la configuración exacta de cada módulo exportado.

En la configuración se ha puesto por defecto que se use como usuario local **nobody**, y que se haga bajo un entorno chroot por seguridad. Esto hace que por defecto cuando nos conectemos a un módulo lo hagamos con ese usuario, y no podamos salir de él a otras zonas del servidor. Dentro de cada módulo se indica la carpeta a la que apunta, se dice que únicamente se pueden leer archivos, mediante **read only = true**, pero no modificarlos ni crear nuevos. Se dice también que si se solicita un listado de módulos al servidor estos estén ocultos, que se utilice como datos de conexión un **usuario** y **password** está en el archivo **/etc/rsyncd.secrets**, y que únicamente se permitan conexiones desde una IP determinada.

Permisos de archivos de configuración

Por temas de seguridad se debe tener encuenta la restricción de permisos, así como se muestra a continuación:

```
chmod 655 /etc/rsyncd.conf
chmod 600 /etc/rsyncd.secrets
```

El archivo que contiene la clave que permite conectar un cliente a un servidor, también debe disponer de los siguientes atributos.

```
chmod 600 /home/rocio/clave.rsync
```

Opciones básicas

- ✔ **-a** para copiar recursivamente manteniendo privilegios, fecha de creación, permisos, etc.
  - ✔ -r recursivo
  - ✔ -l preservar soft links
  - ✔ -p preservar permisos
  - ✔ -t preservar fecha
  - ✔ -g preservar grupo
- ✔ **-v** para incrementar el nivel de detalle de la operación;
- ✔ **-z** para comprimir los datos, así la transferencia es más rápida;
- ✔ **-n** no transferir solo mostrar lo que hay que hacer
- ✔ **-C** ignorar archivos como lo hace CVS
- ✔ **-u** mantiene archivo destino si existe y es posterior
- ✔ **-b** renombrar archivos destino preexistentes a extensión ~
- ✔ **--stats** imprimir estadísticas al final (solo si se ha puesto también -v)
- ✔ **--delete** borrar archivos en destino si no existen
- ✔ **-R** path relativos (crear rutas completas en el destino)
- ✔ **-e ssh** para usar ssh para copiar los archivos de un servidor a otro.
- ✔ **–delete** elimina del destino los ficheros y directorios eliminados en el origen
- ✔ **–exclude=PATTERN** no copia los ficheros que coinciden con PATTERN

Manejo básico

```
rocio@laboratorio: ~
Archivo  Editar  Ver  Terminal  Solapas  Ayuda
rocio@laboratorio:~$
rocio@laboratorio:~$ rsync 192.168.[redacted]::
presentacion    directorio publico de presentaciones
proyecto        directorio privado de proyectos
rocio@laboratorio:~$
rocio@laboratorio:~$ rsync rsync://192.168.[redacted]
presentacion    directorio publico de presentaciones
proyecto        directorio privado de proyectos
rocio@laboratorio:~$
```

Listado de directorios disponibles en el servidor (mostrando 2 modalidades)

```
rocio@laboratorio: ~
Archivo  Editar  Ver  Terminal  Solapas  Ayuda
rocio@laboratorio:~$
rocio@laboratorio:~$ rsync 192.168.[redacted]::presentacion
drwxr-xr-x        4096 2009/01/08 10:48:18 .
-rw-rw-r--        1487 2008/08/28 16:19:10 presentacion.pdf
-rw-rw-r--      722071 2008/08/28 11:58:45 presentacion_00.pdf
-rw-rw-r--        1282 2008/08/28 17:27:24 presentacion_01.pdf
-rw-rw-r--        1755 2008/08/28 10:43:44 presentacion_02.pdf
rocio@laboratorio:~$
rocio@laboratorio:~$ rsync rsync://192.168.[redacted]/presentacion
drwxr-xr-x        4096 2009/01/08 10:48:18 .
-rw-rw-r--        1487 2008/08/28 16:19:10 presentacion.pdf
-rw-rw-r--      722071 2008/08/28 11:58:45 presentacion_00.pdf
-rw-rw-r--        1282 2008/08/28 17:27:24 presentacion_01.pdf
-rw-rw-r--        1755 2008/08/28 10:43:44 presentacion_02.pdf
rocio@laboratorio:~$
```

Listado del contenido de un directorio en particular (mostrando 2 modalidades)

```
rocio@laboratorio: ~
Archivo  Editar  Ver  Terminal  Solapas  Ayuda
rocio@laboratorio:~$
rocio@laboratorio:~$ rsync anibal@192.168.[redacted]::proyecto
Password:
drwxr-xr-x        4096 2009/01/08 10:44:48 .
-rw-rw-r--        1409 2008/08/28 10:08:30 articulo.rst
-rw-rw-r--      584759 2008/08/28 17:09:54 proyecto_final.pdf
rocio@laboratorio:~$
rocio@laboratorio:~$ rsync rsync://anibal@192.168.[redacted]/proyecto
Password:
drwxr-xr-x        4096 2009/01/08 10:44:48 .
-rw-rw-r--        1409 2008/08/28 10:08:30 articulo.rst
-rw-rw-r--      584759 2008/08/28 17:09:54 proyecto_final.pdf
rocio@laboratorio:~$
```

Listado de directorios disponibles solo con un usuario y contraseña (mostrando 2 modalidades)

```
rocio@laboratorio: ~
Archivo  Editar  Ver  Terminal  Solapas  Ayuda
rocio@laboratorio:~$
rocio@laboratorio:~$ rsync -avh 192.168.[redacted]::presentacion pre_rocio/
receiving file list ... done
./
presentacion.pdf
presentacion_00.pdf
presentacion_01.pdf
presentacion_02.pdf

sent 135 bytes  received 727.03K bytes  1.45M bytes/sec
total size is 726.60K  speedup is 1.00
rocio@laboratorio:~$
```

Sincronización básica

```
rocio@laboratorio: ~
Archivo  Editar  Ver  Terminal  Solapas  Ayuda
rocio@laboratorio:~$
rocio@laboratorio:~$ rsync --progress -avh 192.168.[redacted]::presentacion pre_rocio/
receiving file list ...
5 files to consider
./
presentacion.pdf
       1.49K 100%    1.42MB/s    0:00:00 (xfer#1, to-check=3/5)
presentacion_00.pdf
     722.07K 100%    2.39MB/s    0:00:00 (xfer#2, to-check=2/5)
presentacion_01.pdf
       1.28K 100%    4.35kB/s    0:00:00 (xfer#3, to-check=1/5)
presentacion_02.pdf
       1.75K 100%    5.95kB/s    0:00:00 (xfer#4, to-check=0/5)

sent 135 bytes  received 727.03K bytes  484.78K bytes/sec
total size is 726.60K  speedup is 1.00
rocio@laboratorio:~$
```

Sincronización básica, mostrando el progreso de cada transferencia

```
rocio@laboratorio: ~
Archivo  Editar  Ver  Terminal  Solapas  Ayuda
rocio@laboratorio:~$
rocio@laboratorio:~$ rsync rocio@192.168.[redacted]::proyecto
Password:
drwxr-xr-x        4096 2009/01/08 10:44:48 .
-rw-rw-r--        1409 2008/08/28 10:08:30 articulo.rst
-rw-rw-r--      584759 2008/08/28 17:09:54 proyecto_final.pdf
rocio@laboratorio:~$
rocio@laboratorio:~$ rsync --exclude=*.pdf -avzh rocio@192.168.[redacted]::proyecto proy_rocio/
Password:
receiving file list ... done
./
articulo.rst

sent 105 bytes  received 1.28K bytes  252.55 bytes/sec
total size is 1.41K  speedup is 1.01
rocio@laboratorio:~$
```

Sincronización básica, incluyendo filtros de exclusión



```
rocio@laboratorio: ~
Archivo  Editar  Ver  Terminal  Solapas  Ayuda
rocio@laboratorio:~$
rocio@laboratorio:~$ rsync --stats -avh 192.168.[illegible]::presentacion pre_rocio/
receiving file list ... done
./
presentacion.pdf
presentacion_00.pdf
presentacion_01.pdf
presentacion_02.pdf

Number of files: 5
Number of files transferred: 4
Total file size: 726.60K bytes
Total transferred file size: 726.60K bytes
Literal data: 726.60K bytes
Matched data: 0 bytes
File list size: 116
File list generation time: 0.001 seconds
File list transfer time: 0.000 seconds
Total bytes sent: 135
Total bytes received: 727.03K

sent 135 bytes  received 727.03K bytes  1.45M bytes/sec
total size is 726.60K  speedup is 1.00
rocio@laboratorio:~$
```

Sincronización básica, mostrando las estadísticas de la transferencia

```
rocio@laboratorio: ~
Archivo  Editar  Ver  Terminal  Solapas  Ayuda
rocio@laboratorio:~$
rocio@laboratorio:~$ rsync --password-file=clave.rsync -avzh rocio@192.168.[illegible]::proyecto proy_rocio/
receiving file list ... done
./
articulo.rst
proyecto_final.pdf

sent 116 bytes  received 534.41K bytes  1.07M bytes/sec
total size is 586.17K  speedup is 1.10
rocio@laboratorio:~$
```

Sincronización básica, incluyendo la clave de acceso en un archivo

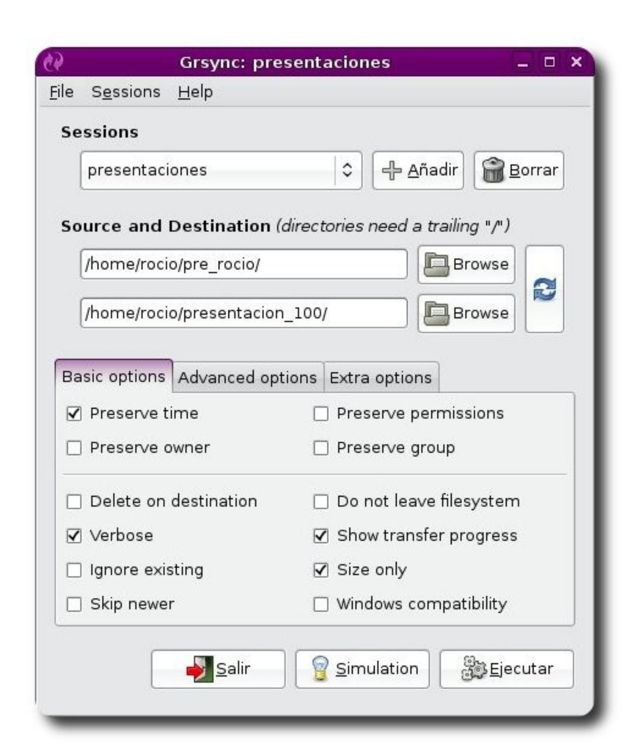
Front-ends

Muchos usuarios no están acostumbrados a la línea de comandos, y prefieren casi siempre hacer uso de un frontend, que les facilite su trabajo, bueno es cuestión de opiniones. En el caso de rsync éste posee algunos frontends como:

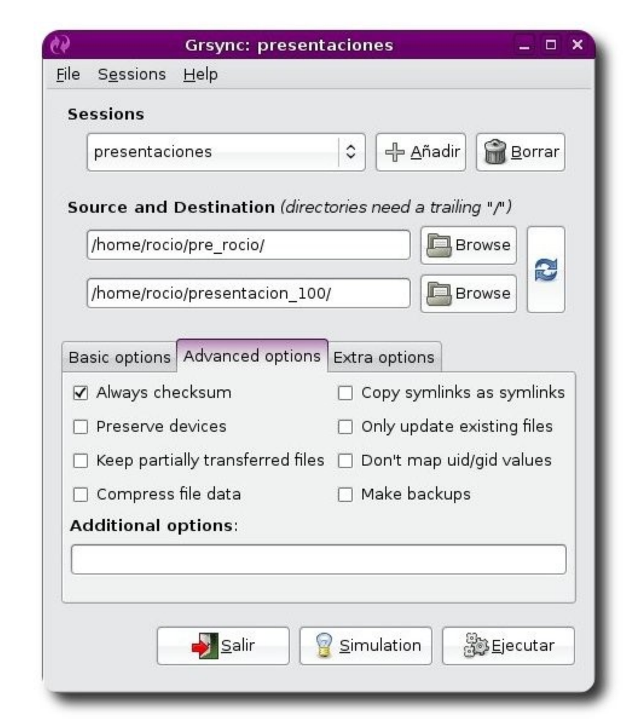
- ✔ **Grsync**: http://www.opbyte.it/grsync/
- ✔ **Qsync**: http://transamrit.net/projects/qsync/
- ✔ **Zynk**: http://hanez.org/zynk.html

y algunos otros, lastimosamente no todos éstos están presentes en los repositorios de las diferentes distribuciones, aspecto que se convierte en un obstáculo para algunos, ya que deben ser instalados desde los fuentes.

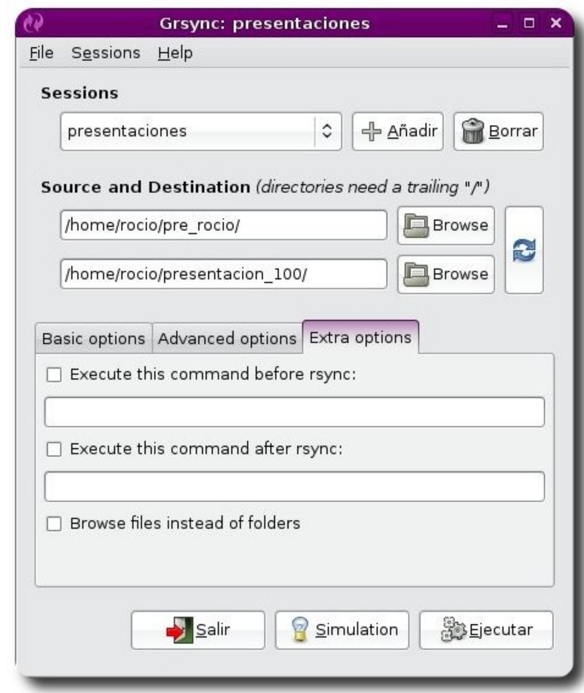
Grsync es un frontend presente en la mayoría de las distribuciones más utilizadas como Debian, Ubuntu, Fedora, CentOS,...Es así que mostraremos algunas de sus características en los siguientes gráficos.



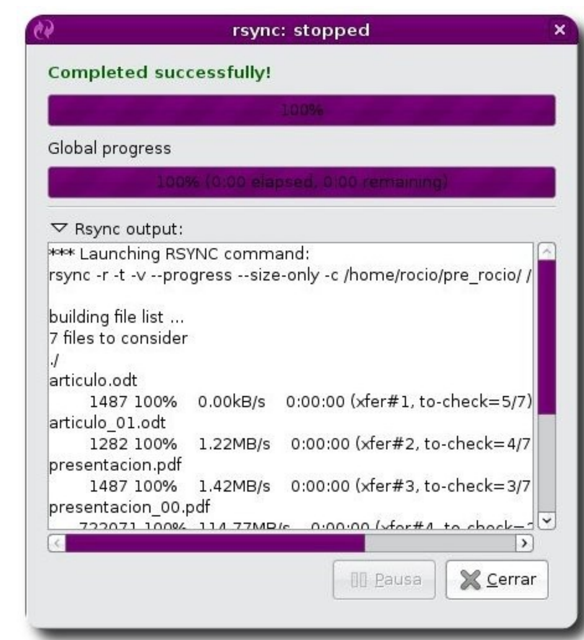
Opciones básicas



Opciones avanzadas



Opciones extras



La ejecución de rsync, muestra las opciones utilizadas

Alternativa a los frontends

En especial para el uso de rsync, aplicado a un entorno de producción, es conveniente desarrollarse unos scripts, y éstos pueden ser utilizados para incluirlos dentro un cron y de esta forma tener todas las tareas automatizadas.

Aquí mostramos unos breves ejemplos:

Script, que realiza el backup de un origen a un destino tomando en cuenta el adto de la fecha que se realiza.

```
#! /bin/bash
FECHA=$(date "+%Y%m%d-%H%M")
ORIGEN="/dir/origen"
DESTINO="usuario@host:/dir/destino"
if rsync --delete -avb --backup-dir="BACKUP-$FECHA" $ORIGEN $DESTINO
then
    echo "Copia exitosa"
else
    echo "Copia Fallida"
fi
```

Script que traslada desde un repositorio los datos de un sitio web, hacia el servidor web donde se publicará.

```
#! /bin/bash
SERVIDOR="100.20.3.4"
ARCHIVOS="data src"
if rsync -rtlv --delete --delete-excluded --exclude=CVS $ARCHIVOS
root@$SERVIDOR:/var/www/html
then
    echo "Publicacion correcta"
else
    echo "Publicacion Fallida"
fi
```

Agradecimiento

Deseo expresar mi agradecimiento al **Ing. Esteban Saavedra** (director de la **Revista Atix**), quién continuamente va motivando y asesorando el desarrollo de trabajos en temas de Software Libre dentro de la universidad y fuera de esta.

Referencias

[1] http://rsync.samba.org/

Autor



Rocio Figueroa
Estudiante de Ingeniería en Telecomunicaciones
rocios.figueroa@gmail.com


Celebremos
Nuestro
Primer
Aniversario

SALUTACIÓN PRIMER ANIVERSARIO

Se dice que "**Cuando más duro sea el camino que recorrer, será más grande la satisfacción de haberlo conseguido**"; es así que la experiencia nos enseña tres cualidades que deberíamos pretender tener todas las personas:

- ✔ **Ser emprendedore**s, ya que el emprender nuevas actividades y proyectos, nos conducirá a nuevos rumbos, a nuevos conocimientos, a nuevas experiencias, elementos que alimentan nuestra mente y nuestro espíritu.
- ✔ **Ser perseverantes**, por que la perseverancia hará que podamos cumplir nuestros objetivos y llegar a nuestras metas
- ✔ **Mantener disciplina**, un elemento que hará que nuestra actividad y nuestros proyectos sean solidos, sostenibles y perduren en el tiempo

Estoy seguro que estas tres cualidades las tratamos de emplear desde el momento que emprendimos la difícil tarea de publicar una revista, más aún teniendo por objetivo principal el de contribuir a la difusión y a promover el uso del conocimiento, las tecnologías y el software libre; ahora después de un año, podemos asegurar que contribuimos y aprendimos muchísimas cosas, a lo mejor más cosas de las que supusimos aprender o contribuir, pero nos encontramos satisfechos de haber llegado a un año lleno de experiencias, que nos permitirán a la postre mejorar nuestro trabajo y la labor que decidimos emprender.

Durante este año, no solo cruzamos fronteras físicas, sino también fronteras sociales, políticas y éticas, por que abrimos nuestras paginas, para que por medio de ellas se pueda contribuir al conocimiento de los demás; tratamos de poner en practica el significado de Atix, por que seguros estamos que todas las comunidades de américa latina **LO SABEN, LO INTENTAN, LO PUEDEN** y **LO LOGRAN**, siempre que se lo propongan.

Agradecer, a todas las personas que nos brindaron su apoyo, en los momentos que lo necesitamos, a nuestros lectores por cedernos parte de su tiempo para leer nuestra revista, a nuestros autores por compartir sus conocimientos y experiencias, y también a todas a las personas que nos enviaron sus mensajes de libertad, brindándonos palabras de aliento que de seguro sirvieron y servirán de mucho, en nuestro continuo caminar en busca de nuestra libertad.

Gracias y larga vida para nuestra Revista Atix.



Esteban Saavedra López
Director y Coordinador General.
Revista ATIX



Felicitaciones por su primer año a todo el equipo humano de la revista Atix, que número tras número nos entrega buen contenido, que todos los amantes del software libre podemos disfrutar.



Estamos dentro de un proceso de cambio tecnológico, y tener claro el horizonte en medio de una neblina es un trabajo de cada día. La revista Atix cumple con la función de guía ante la tiniebla tecnológica que cubre nuestro país.
Felicito a todos los que hacen posible este emprendimiento en pro del software libre y les deseo éxitos en sus labores.



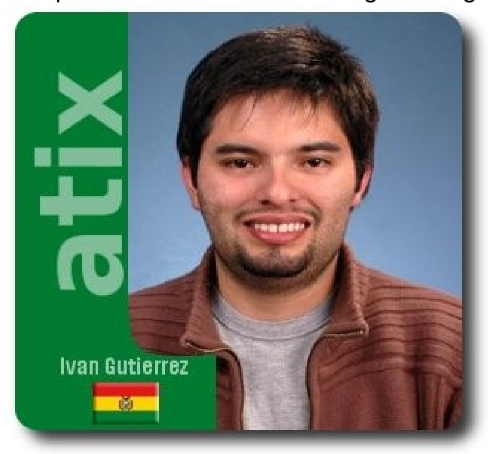
Llevar adelante un proyecto implica poner mucho cariño, tiempo y esfuerzo, para sacarlo adelante. Como comunidad estamos consientes que no es fácil hacer esto y requiere tener constancia y calidad en el trabajo realizado.
Realizar este trabajo por un año, es el reflejo de la calidad y dedicación que pone cada una de las personas integrantes de Atix. Mis felicitaciones y desearles muchos años más de vida y que cumplan sus metas, en este nuestro mundo de libertad.



Es motivo de orgullo para todos nosotros, contar con una revista boliviana que sale de los contenidos tradicionales de una publicación informática y nos ofrece contenidos innovadores que cooperan y promueven la difusión de Software Libre traspasando hábilmente nuestras fronteras.
Felicitaciones y gracias a la Revista ATIX por proveernos y compartir un año de conocimiento y experiencias de la única revista Boliviana 100% de Software Libre. ¡Éxitos!



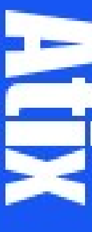
Es un honor escribir estas cortas palabras de felicitación con motivo de su primer aniversario de la revista ATIX, al mismo tiempo reconocer el esfuerzo dedicado al difundir y promover el uso de software libre a través de los artículos publicados por personas reconocidas del medio e involucradas con el uso del mismo, particularmente me enorgullece saber que la revista "ATIX" cruzo las fronteras obteniendo a través del tiempo, mayor trascendencia y aceptación por los usuario y lectores involucrados con el software libre.
Valga la redundancia FELICIDADES Revista ATIX y a quienes componen el grupo de trabajo y uno especial para Esteban Saavedra un gran amigo.



Felicidades a ATIX, por este primer año. Un gran reconocimiento a todas las personas que están detrás de la publicación, recuerden que hacen un gran trabajo por la Filosofía, Tecnología, Software y Conocimiento Libre de Bolivia, Latinoamerica y el Mundo en General. Sigan adelante y que cumplan muchos, pero muchos años más.

atix
Daniel Viñar

En estos días, el numero 1 de la revista ATIX completa su primera vuelta alrededor del sol. Entre tanto, siete números más pautaron la órbita de nuestro planeta azul, con una regularidad digna de los elementos. Cabe saludar ante todo, la seriedad y el profesionalismo que ha demostrado el equipo de la revista ATIX.
Es por eso, sin duda, que ha logrado más de quinientas páginas de contribuciones sobre software libre, abundantes y sabrosas, de calidad, eclécticas y sumamente diversas. Hay artículos tanto para especialistas, como para cualquier persona que quiera conocer y utilizar software libre. Participa gente del mundo entero. Abarca desde el arte y el grafismo hasta el desarrollo, pasando por la educación o el cine. Aporta valiosas explicaciones y enseñanzas técnicas, pero también noticias, información, reflexiones, entrevistas y hasta cultura o turismo. Diversidad de género, también con una participación femenina poco común en el mundo de la informática libre.
Y como si fuera poco la revista se destaca por su calidad gráfica y su belleza, temas a los que nuestro mundo "geek" no siempre es atento.
Todo esto está a disposición nuestra, bajo una licencia verdaderamente libre: CC-BY-SA. Podemos hacer todo lo que queramos con la riqueza que contiene la revista: utilizarla, estudiarla, mejorarla y difundirla, bajo la única condición de citar la fuente y preservar la licencia.
Quizás deberíamos tomar estos derechos como un reto ético y político: ¿qué podemos hacer, todas y cada uno, para difundir el conocimiento libremente compartido, para explorarlo y aumentarlo? ¿Y cómo podemos limitar el alcance en nuestra civilización de quienes pretenden generalizar el secreto y la dependencia tecnológica?
Por eso, multipliquemos los ejemplares de la revista ATIX, démosla a conocer enviándola, imprimiéndola ¡e incluso vendiéndola! Utilizémos y ré-utilizémos todo lo que nos da. Contribuyámos entendiéndolo, criticándolo, extendiéndolo, mejorándolo, retomándolo.
¡Feliz cumpleaños y larga vida, querido ATIX! Te debemos un pedacito de nuestra libertad.





¡Que alegría que ATIX cumpla ya su primer año! Para los que amamos el software libre y nos encanta curiosear en información técnica - sin dejar de lado el compromiso ético que lo sustenta - es muy adecuada la calidad de las notas y la selección variada de temas de la revista, que se ha ido convirtiendo en una importante referencia entre las publicaciones de software libre en nuestro idioma.
Les mando un fuerte abrazo a todos los amigos que realizan Atix, les deseo muchos años más de trabajo, sigan así! Muchas gracias por compartir siempre tan buen material.



Mi opinión es que ATIX, siendo una de las pocas revistas digitales sobre Software Libre de Latinoamérica, ha seguido una línea editorial clara y no hace sino sumar adhesiones desde el primer día, gracias a una excelente selección editorial y a no tener el estilo pretencioso de otras publicaciones de corte más comercial tradicional. Una de las mejores iniciativas regionales que he visto surgir en los últimos 2 años. Por eso me alegra mucho que el proyecto prospere sólidamente.
Un abrazo afectuoso!



Estimados amigos de Atix

Desde Argentina quiero brindar con ustedes y agradecerles por todo el esfuerzo y dedicación que le ofrecen a la comunidad para llevar a cabo la edición de esta gran revista.
Un fraternal abrazo.



Desde Buenos Aires, quiero enviar mis saludos a la gente de Atix, que cumple su primer año de existencia. Generalmente en las salutaciones se desea “otro año más de vida” y felicitaciones, lo cual es síntoma de que se está trabajando más que bien en el tema.
Pero lo más importante para Bolivia es que cuenta con una revista seria, comprometida con la ética del software libre y por sobre todo, comunica con rigor técnico y moral lo mejor que el software libre Boliviano tiene para dar: Buenas Personas comprometidas con la sociedad Boliviana y Latinoamericana.
Mis saludos para mis hermanos Bolivianos de esta gran Nación Latinoamericana.



Congratulo a la comunidad ATIX por el cumpleaños de la revista que, a pesar de tan joven, ya es una referencia regional y internacional, no sólo por la calidad y relevancia de los artículos técnicos, sino también por el cuidado con el diseño y el espacio para cultura, humor, novedades y la valorización de sus orígenes, en la cultura tradicional y el incentivo al turismo en el país.
Sin embargo, me preocupa la atención a los aspectos tecnológicos del Software Libre, haciendo parecer que el Movimiento Software Libre no es un movimiento social, moral, ético y político, sino tecnológico. Así que invito la comunidad a expandir su atención a las cuestiones más importantes de nuestro movimiento, y llevar a su público más elementos de la filosofía de conciencia social, para la libertad y el respeto al prójimo.
Por supuesto, me ofrezco para contribuir acerca de estos temas, que espero podrían hacer de la Revista ATIX un medio aún más efectivo para incentivar el uso y la cultura del Software Libre, sino también para hacerla más interesante para un público más amplio, y un ejemplo, más completo aún, a ser destacado, recomendado y imitado.
¡Un saludo a la comunidad de Software Libre de Oruro y las de toda la América Latina, y larga vida, con muchos logros, para la Revista ATIX!



A Atix é a mostra da capacidade do povo boliviano. Uma revista bem organizada, um sítio bastante funcional, e conteúdos importantes, atuais e contextualizados. Mas, principalmente, o pessoal da Atix tem uma boa vontade exemplar, de todos os povos da América Latina. Atix e Latinoware são duas instituições irmãs, no meio de tantos irmãos, que nem podemos contar.
Parabéns Bolívia por ter a Atix. Viva o primeiro de muitos aniversários.



En los últimos cien años la América Latina y África han tenido una expoliación por parte de las gran empresas de los países ricos en una escala nunca vista en la historia.
El uso de la tecnología inalámbrica es una oportunidad de mudar esta historia.
El software libre es una muy buena opción de llevar tecnología y oportunidad de empleo hasta nuestros pueblos más pobres en los más distantes rincones. Sin la pérdida de nuestros recursos naturales y con la preservación de las dignidades nacionales.
Es por esto que me gustaría felicitar al equipo de Atix por este un año de muy bueno trabajo en la divulgación del software libre en Bolívia y en toda la américa latina.



Se prende la primera vela para Atix
Con mucho agrado celebro este primer aniversario de Atix, la revista de software libre que se lee en toda latinoamérica, y seguramente en otros lugares del mundo también, en donde haya alguien que se interese por las tecnologías libres y lea español, lo cual no es difícil de encontrar.

En un mundo como éste, lo que sí es difícil de conseguir es que un grupo de personas trabajen metódica y concienzudamente en un proyecto por el cual no se cobra un centavo, sino más bien en donde los beneficios obtenidos de donar el tiempo de trabajo son otros, como una tranquilidad de conciencia por saber que se hace lo correcto, que se genera conocimiento colectivo y se comparte en libertad, lo cual naturalmente constituye un aporte a nuestra sociedad. La primera vez que leí la revista, me gustó muchísimo, porque de inmediato supe que no era una publicación más de tecnología, sino que una que se preocupa de entregar valores, por eso es que se define como de software libre y no de otro tipo. Además, integra al entorno las bondades del maravilloso país que es Bolivia, dándonos el mensaje que las tecnologías y el conocimiento deben estar al servicio de nuestra sociedad y sus personas, y no solo como objeto de uso. Es por esto que valoro muchísimo la labor que está desarrollando el equipo de trabajo de Atix, les envío un fuerte abrazo desde Chile, y los animo a seguir con fuerzas esta invaluable labor.



La capacidad de pensar, ha brindado a los seres humanos libertad; y junto a ésta, la capacidad de aprender, siendo lo aprendido, una expresión de nuestra propia libertad. En consecuencia, el intercambio y la apropiación del conocimiento son para el hombre, un notable motor de su existencia.

Usar, distribuir, copiar, aprender es ser libre Use software libre. En su primer aniversario, un gran abrazo y los mejores deseos de libertad y conocimiento.



¡Saludos a todos los lectores y miembros de la revista Atix!



Vaya desde aquí mi más afectuosa enhorabuena a una revista que en menos de un año ha sabido, desde prácticamente la nada y sin casi presupuesto alguno, sólo con el sudor, la dedicación, el bien hacer de sus progenitores y siguiendo el paradigma digital, abrirse un hueco de calidad en el concurrido mundo editorial del software libre. Sean muchos más los años de éxitos que os depare el futuro para goce de quien se deleita leyéndoos como es mi caso. ¡Gracias!



Desde Ecuador quería enviar un grato saludo y una felicitación para le revista ATIX por el gran trabajo que desempeña promocionando el software libre. La revista ATIX es mucho más que un proyecto Boliviano, es una de las herramientas más importantes por la que usuarios novatos y avanzados podemos expandir nuestro conocimiento en el mundo del software libre. Felicitaciones y esperamos que cumplan muchos más años para seguirlos leyendo.



¡Hermanos bolivianos!
Me entero con mucho gusto del primer aniversario de la revista Atix. La comunidad de Software Libre en Bolivia es una de las mejor organizadas y más exitosas de las que conozco, y el que esta revista se establezca y se mantenga es una prueba más del empeño que han puesto en la profesionalización del Software Libre en su país.
¡Muchas felicidades, adelante con estos importantes esfuerzos!



Cada edición de la Revista de Software Libre Atix, es una oportunidad concreta y real para conocer los beneficios sobre el uso y desarrollo del software libre en este mundo gloabilizado, como aporte informativo a la Comunidad Latinoamericana de Software Libre, quiero expresar mis cordiales saludos por el Primer Gran Aniversario, al mismo tiempo hago votos para que continúen este gran esfuerzo



Estimados amigos de la Revista ATIX,
Las personas que participan de comunidades de software libre y del código abierto se mueven por una gran variedad de motivaciones. Sin embargo, una de las más frecuentes es la de compartir con otros su pasión por la tecnología y sus conocimientos. Desde esta perspectiva, elaborar una publicación periódica como la revista que Uds. Elaboran no es, en esencia, muy distinto de desarrollar una pieza de software y publicarla bajo una licencia libre. Además, Uds. llevan en su esfuerzo el espíritu mismo del hacker: les apasiona el conocimiento, descubrir o aprender nuevas cosas y entender el funcionamiento de éstas.
Me gustaría poder llamarlos "**hackers de la tinta y el papel**" pero creo que no será posible ya que uno de sus "**hacks**" es justamente sacarle el provecho a los medios digitales para hacer llegar su valiosa contribución al conocimiento de estos temas que nos apasionan tanto utilizando el idioma castellano y pensando en los lectores de Latinoamérica.
Es así que en nombre del Grupo de Usuarios de Linux del Perú, como uno de sus coordinadores, me complace mucho saludarlos en este aniversario y desearles una larga vida como proyecto editorial ya que cualquier esfuerzo por difundir el software libre que tantos beneficios trae a nuestras economías locales y nuestro propio desarrollo profesional y personal. Desde este país hermano reciban un fraternal abrazo allá en Bolivia pero también en cada uno de los rincones del mundo a donde llega una copia de la revista, ya que la familia ATIX la integramos todos: los que escriben y los que leen, los que investigan y prueban, los que programan y despliegan, todos haciendo un aporte personal a este gran movimiento que es el Software Libre.

Estimados Amigos de ATIX:
Nuestra región, nuevamente cruza una etapa transcendente en que disponemos la decisión de mantener nuestra libertad e independencia, pero visto de otra arista, el de la Tecnología de la Información y Comunicación.
El Software Libre y el OpenSource son fuertes catalizadores para el reecuentro de la soberanía de nuestras libertades, estas nuevas herramientas son formidables impulsores de desarrollo y de depuración para nuestras naciones, no desde un punto político sino desde un punto filosófico en donde todos tenemos el acceso a la información, el poder de decisión y de acción.
Por tanto iniciativas como ATIX son bienvenidas en nuestras comunidades porque rompemos esquemas, prejuicios, y fronteras.
Desde Paraguay deseamos grandes augurios y éxitos para ATIX. Feliz Aniversario.



El Software Libre en el Paraguay está ya posicionado, va ganando espacios importantes en la opinión de la gente y especialmente en el Estado Paraguayo que es nuestro objetivo en este periodo del Presidente Fernando Lugo, quien ha apostado a la Libertades que el Software Libre nos da. Reconocemos que falta



Mis mayores felicitaciones al cumplir vuestro primer aniversario. He incursionado alguna vez en el mundo editorial, una de ellas con una revista multimedia en CD-ROM y algo me quedo claro, sacar el primer número de una revista es muy fácil, lo difícil es sacar el número dos. Ustedes ya cumplen un año y realizan un aporte muy importante a la comunidad de Software Libre. Así que felicitaciones y feliz cumpleaños. La torta más que una velita por año debería incluir una velita por número.



Saludos desde Venezuela. Es un honor presenciar el primero de lo que todos deseamos sean muchos aniversarios en este esfuerzo por la libertad. Mi admiración y mis respetos para todas aquellas personas que han hecho posible este frente por la liberación del conocimiento.

aun sortear varios obstáculos que son muy comunes en todos los países de Latinoamerica, creo que compartimos muchos problemas concernientes a la difusión y el entendimiento de la Filosofía del Software Libre, pero lo bueno hay que resaltarlo y eso es que estamos dando pasos firmes y concretando metas.
Desde la Comunidad de Software Libre del Paraguay quiero agradecerle a Esteban Saavedra por la oportunidad que nos da, de ser partícipes de la difusión plena en la región y aportando nuestro granito de arena a la gran causa que habría iniciado nuestro gran sostén ideológico RMS.



La excelencia se mide en el esfuerzo y amor con el que hacemos las cosas. Tener fe y perseverancia son las claves del éxito en cualquier camino que se emprende. A ustedes, felicitaciones por ser un ejemplo para los que defendemos una ideología que busca la difusión masiva de la libertad como forma de vida. Felicitaciones!


Willay news
Información Actual

Willay news

World Wide Web 2009



En los 20 años de Aniversario de la Web Madrid cede del Congreso **World Wide Web 2009**

Han pasado 20 años de la creación de la web y Madrid será la cede del XVIII congreso **World Wide Web 2009**.

Nuevamente España será la cede de uno de los eventos más importante a nivel mundial. El **World Wide Web 2009** se llevará a cabo en Madrid del 20 al 24 de abril y contará con la asistencia de personajes importantes. Se tendrá la presencia del inventor de la Web Tim Berners Lee, que hace 20 años diseñó el lenguaje base para la creación de páginas web el "HTML" y también se contará con la presencia de Vinton Cerf conocido como el "Padre del Internet", que diseñó un conjunto de protocolos conocidos como TCP/IP y por su trayectoria académica y de investigación Tim Berners Lee y Vinton Cerf serán nombrados doctor honoris causa por la Universidad Politécnica de Madrid, a instancia de la Escuela Técnica Superior de Ingenieros de Telecomunicación.

Junto a la presencia del inventor de la web y el padre del internet se tendrá la presencia de representantes de algunas empresas de renombre como Apache, Mozilla, Google entre otros.



El Congreso estará compuesto por una serie de paneles, mesas redondas donde se tratará sobre todo lo referente a la Web, por ejemplo las redes sociales, la web 2.0, los buscadores y otros.

WhyFLOSS 2009



El 21 de mayo se llevará a cabo la Conferencia **WhyFloss 2009** en el campus de Leganés de la Universidad Carlos III de Madrid (España).

WhyFloss es una unidad de neurowork, dedicada a promover el uso de tecnologías, free, libre y open source Software, para ello realiza una serie de eventos, conferencias donde se publica la promoción y la difusión del software libre y de código abierto en todos los ámbitos de la sociedad y sobre todo los éxitos que se ha tenido en distintos sectores que utilizan estas herramientas. "Así surgió la idea de llevar a cabo una conferencia que mostrará gracias a ejemplos concretos de éxito las razones para usar el software libre y de fuentes abiertas en diferentes lugares del mundo", comenta Alejandro Sánchez, director de desarrollo de negocio de Neurowork, y responsable de WhyFLOSS en Europa. Para más información visita la página **www.whyfloss.com**.

Autor

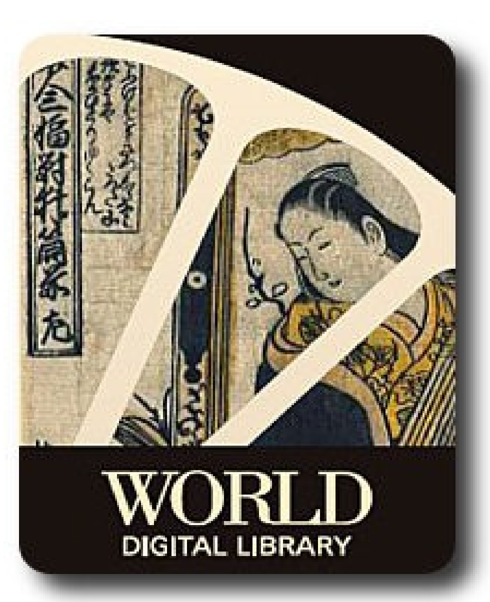


Ivonne Karina Menacho Mollo
Titulada de la carrera de Ingeniería de Sistemas e Informática (F.N.I.)
Conductora programa radial “**Doble Clic**”
ivonnekarina2003@hotmail.com

Willay news

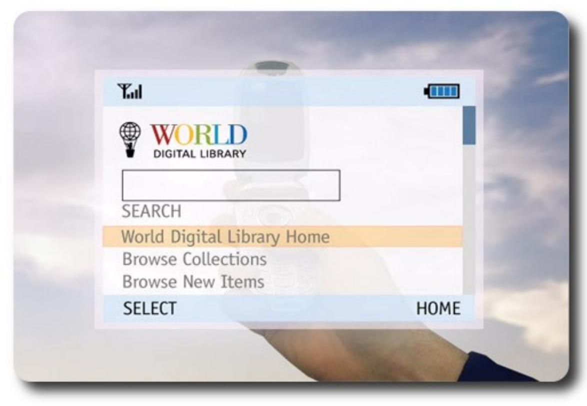
Biblioteca Digital Mundial

Los secretos culturales más espectaculares de todo el mundo ya no estarán guardados, de hecho ahora estarán a disposición de cualquier persona en el mundo de manera gratuita e ilimitada.



El proyecto que empezó como una loca pero espectacular idea por James Billington se hace realidad, este 21 de abril la UNESCO junto a otras 32 instituciones presentaran **World Digital Library** (Biblioteca Digital Mundial) una pagina Web llena de toda la historia cultural de nuestro mundo, arte, ciencia, libros raros, partituras musicales, dibujos arquitectónicos, fotografías, audio y vídeos que seguramente nos impresionaran con su uso tan interactivo e intuitivo que caracterizará a esta pagina.

Este sitio de sabiduría denominado: "**La Catedral Mundial on line**" tiene el objetivo de acercar a las diferentes culturas, impulsar una mejor comprensión entre ellas, proporcionar usos educacionales y contribuir a la investigación académica.



Word Digital Library proyecto millonario financiado por donaciones privadas entre varias Google y Microsoft, contará con una amplia red de bibliotecas de muchos países, estará disponible en siete idiomas: ingles, español, francés, portugués, chino, ruso árabe.

Aunque por ahora algunos países en vías de desarrollo no estén incluidos en el proyecto, pues seguramente los lectores de esta nueva oportunidad que se brinda, para el conocimiento de nuestras culturas, incentivaran a que se rompan esas limitaciones.

Rastrea tu computadora robada con: "Prey"

Solo supongamos… perder toda tu información, el último proyecto de la compañía o del colegio, los programas que te costaron conseguir, la música que te llevó horas y horas clasificar y ordenar, las fotos de la ultima vacación, etc. todo eso porque te robaron tu tan preciada laptop. Por supuesto todos haríamos lo imposible para recuperarla.



Este fue el caso de Tómas Pollak, un periodista, que inspirado en el robo de su computador, se dio el trabajo de desarrollar un programa que te permite rastrear tu computadora robada.

Este programa que fue bautizada con el nombre de "**Prey**" (presa), te permite recolectar un lote de información de tu computadora desde Información de Red (p.e. la dirección IP pública y privada de donde esté conectado el PC), Información interna del PC (p.e. cuanto tiempo lleva encendido el aparato) hasta Información del Ladrón (una foto del impostor si tiene webcam, un pantallazo del escritorio para ver que esta haciendo) y te la envía a un correo electrónico que hayas definido previamente, ah! y si usas Linux hasta puedes generar un cuadro de dialogo con el impostor.

Según Pollak "**Prey**" debería correr en cualquier variante de NIX (Linux, Mac, etc) por ahora se ha probado en Ubuntu Intrepid y en Mac Os Leopard, y espera que con ayuda de muchos interesados se haga mejoras en este tan interesante programa, ya que fue liberado bajo la licencia GPL.3 con un añadido SRTCRMCUC que significa "**Si Recuperas Tu Computador Robado Me Compras Una Cerveza**"



Seguramente con el éxito que tendrá "Prey" Tómas Pollak deberá ir pensando en una bodega donde ir guardando tanta cerveza.

Autor



Marcia Estrella Velasquez Aguilar
Egresada de la carrera de Ingeniería de Sistemas (F.N.I.)
mevaguerreradelaluz@gmail.com

Willay news

Robots y equipos de auxilio

El personal que trabaja en emergencias y seguridad, generalmente necesita estar siempre con lo último en tecnología de comunicaciones, tanto conectividad para móviles, wireless, radio o GPS.

Una universidad de Alemania ha desarrollado un pequeño robot volador que sumará ayuda en casos de emergencia. Por ejemplo, podrá llevar volando conectividad móvil y Wi-Fi a zonas de desastre donde los afectados no puedan moverse o estén heridos en el piso, por ejemplo.



Cada uno de estos robots tienen un costo de $380 dólares, sin embargo la batería no está incluida y es su parte más cara, cuesta $1200 dólares y tan sólo dura para 20 minutos de vuelo.,

Los robots ya vienen siendo usados para asistir en misiones de rescate y desastre cuando se encuentran víctimas atrapadas en derrumbes, terremotos y demás.

Sin embargo este robot pretende ayudar más eficientemente que otros de su categoría, ya que se trata de un robot que puede llegar hasta donde están las víctimas, cargarlas y llevarlas sanas y salvas hasta un punto neutro del desastre.



Este robot se está usando en pruebas en el departamento de bomberos de Yokohama, Japón, allí se ha constatado que este robot puede levantar y trasladar hasta personas de 110 kgs.

Hardware para no videntes

Escribir en un computador, navegar por Internet, jugar o aprender con programas multimedia, leer textos en pantalla o simplemente desplazar el mouse son actividades tan sencillas y hasta rutinarias para la mayoría, que por lo general, desconocemos la realidad de los miles discapacitados que existen en el mundo que necesitan costosas herramientas de hardware y software para desarrollar esta misma clase de actividades.



Lo más grave, es que para una persona discapacitada acceder a esta tecnología es más esencial que para el resto

Es por eso que muchos fabricantes de hardware están poniendo mucho interés en poder desarrollar todo tipo de dispositivos que permitan disminuir las diferencias de acceso a las TICs por parte de los discapacitados.

Un ejemplo es la posibilidad de realizar impresiones en braile, mediante la impresora mostrada a continuación.



Un invento especialmente diseñado para personas no videntes. Se trata de la impresora de etiquetas Braile, ayuda a las personas que no tienen visión a tener una mejor idea de las cosas que los rodean al poder etiquetar objetos por ejemplo en la heladera, estantes, muebles y demás lugares.

Autor

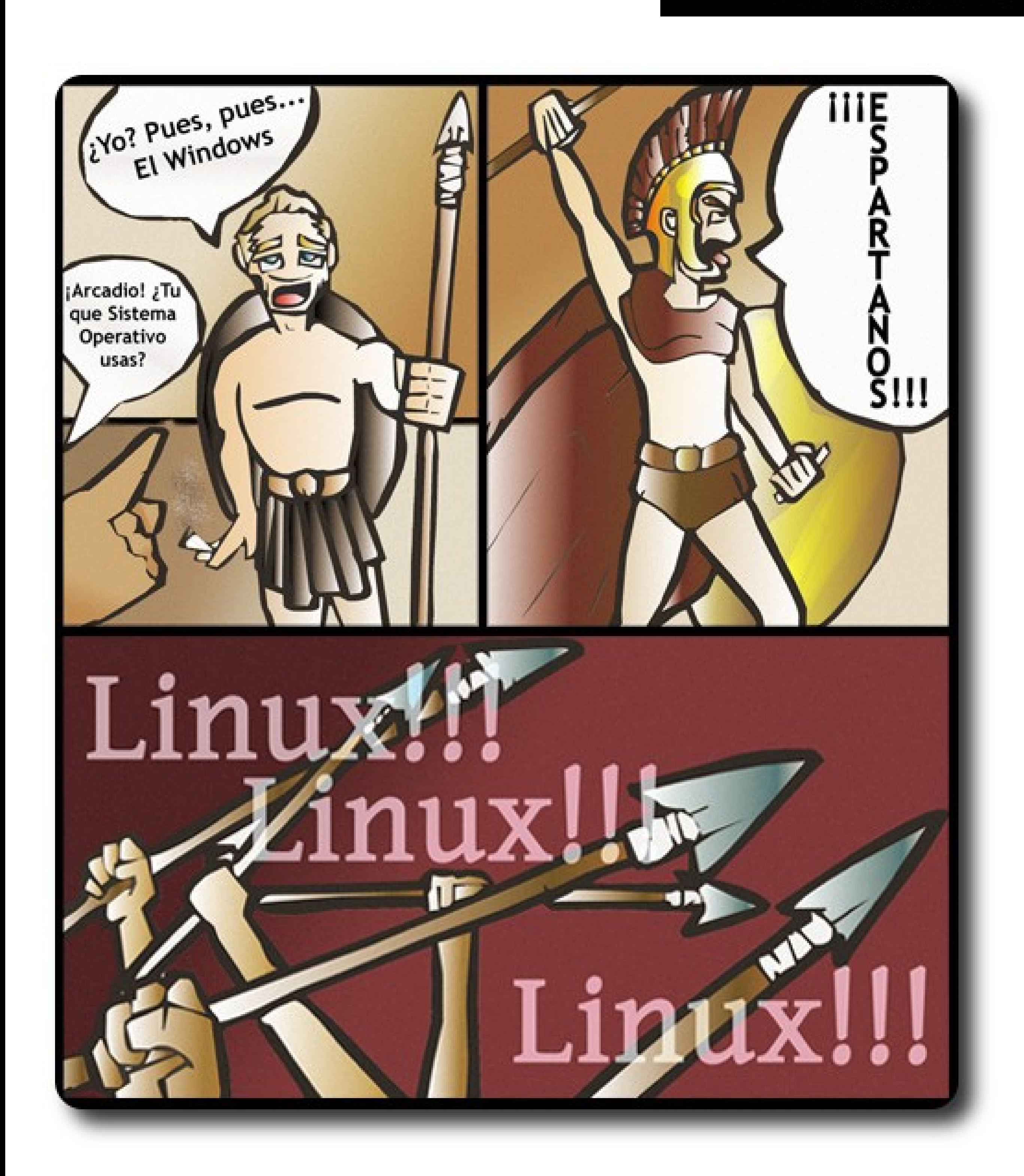


Jenny Saavedra López
Diseño y Edición Revista Atix
jennysaavedra@gmail.com

SOFTWARE LIBRE DEBES UTLIZAR



Comics


¡Arcadio! ¿Tu que Sistema Operativo usas?
¿Yo? Pues, pues... El Windows
¡¡¡ESPARTANOS!!!
Linux!!!
Linux!!!
Linux!!!



Conociendo lo Nuestro

Pando



Amazonía de Pando



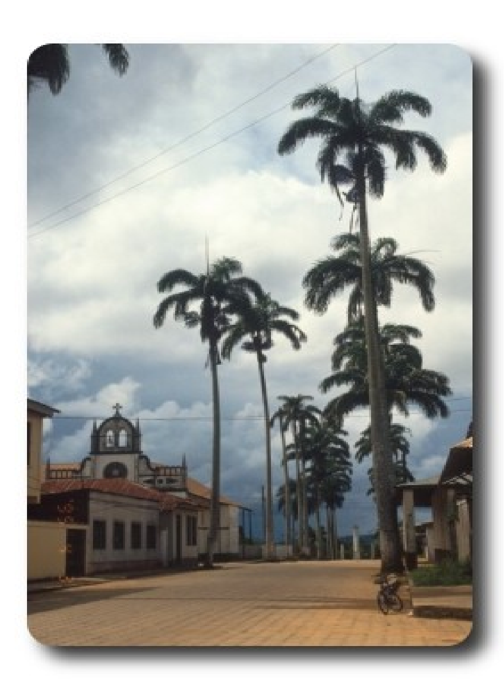
Selva amazónica de Pando



Fauna de la amazonía de Pando



Vista de la ciudad de Cobija - Pando



Campo Ana - Pando



Campo Ana - Pando

Libres para pensar, libres para decidir, libres para crear



Te ofrecemos este espacio para mostrar tu Creatividad

ATIX

La Fuerza del Software Libre

Envíanos tus diseños y creaciones para publicarlos


STAR ATIX

Consultas

Contacto

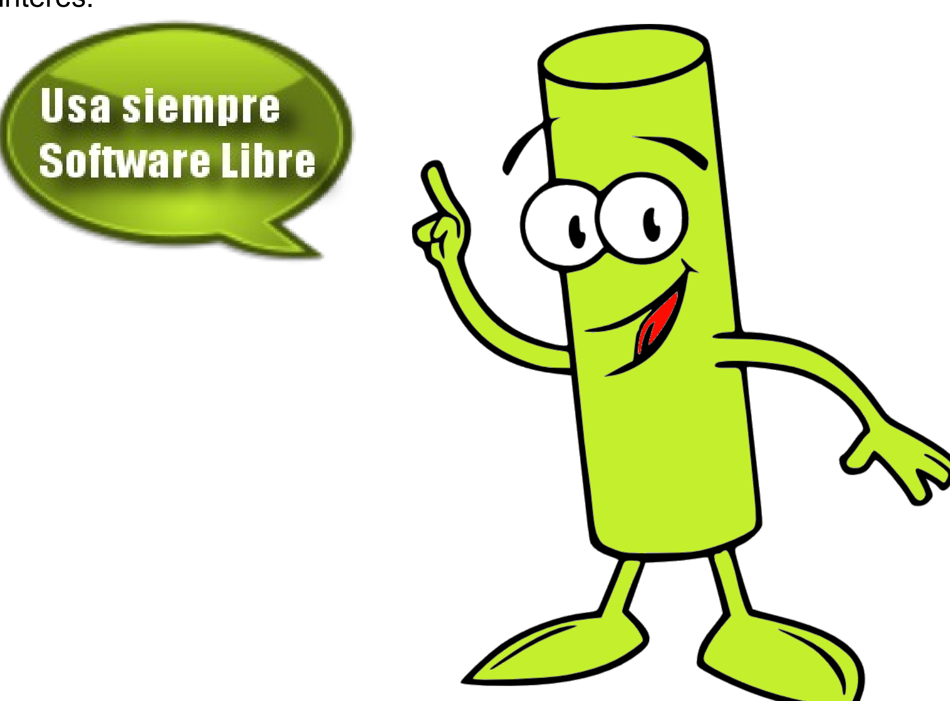
Para solicitar cualquier información, puedes contactar a:

- ✔ Esteban Saavedra López (jesaavedra@opentelematics.org)
- ✔ Jenny Saavedra (jennysaavedra@gmail.com)

Publicación

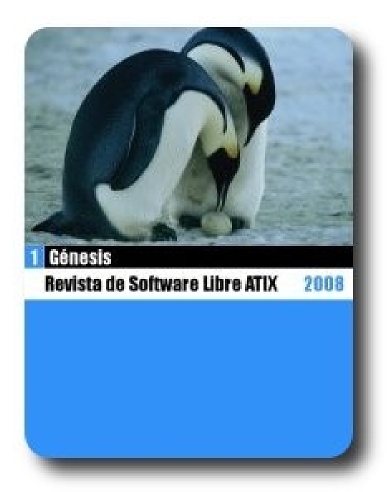
Te invitamos a ser parte de la **Revista ATIX**. La forma de participar puede ser enviándonos:

- ✔ Artículos referidos a áreas como:
  - ✔ Instalación y personalización de Aplicaciones
  - ✔ Scripting
  - ✔ Diseño gráfico
  - ✔ Programación y desarrollo de aplicaciones
  - ✔ Administración de servidores
  - ✔ Seguridad
  - ✔ y cualquier tema enmarcado dentro del uso de Software Libre
- ✔ Trucos y recetas.
- ✔ Noticias.
- ✔ Comics.
- ✔ Links de interés.

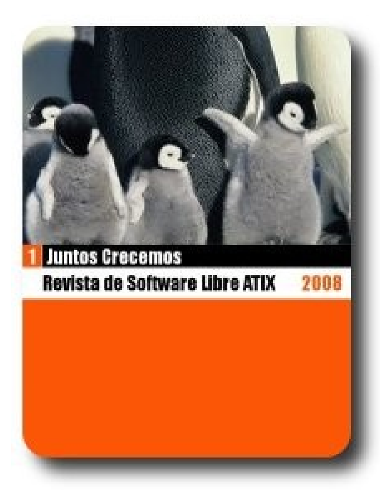


Bitacora

Empezamos a registrar nuestra historia:



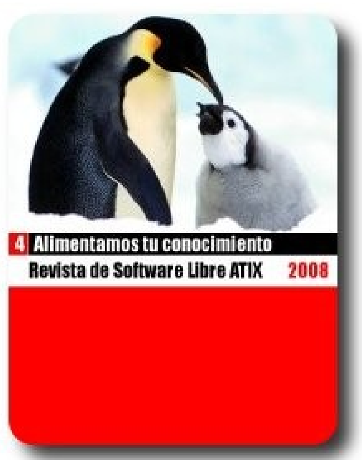
28 de abril del 2008



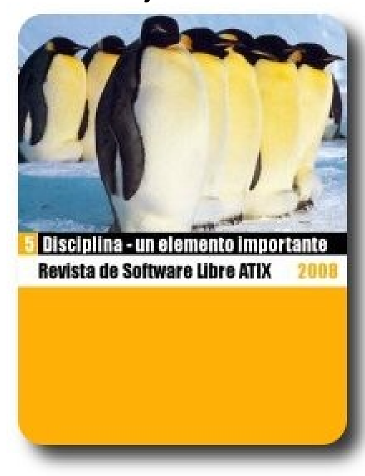
20 de junio del 2008



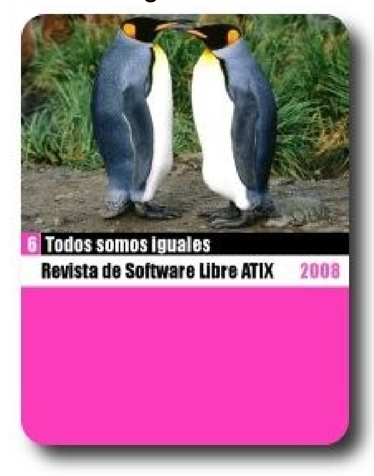
6 de agosto del 2008



17 de septiembre del 2008



10 de noviembre del 2008



17 de diciembre del 2008



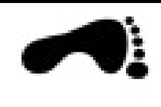
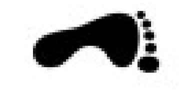
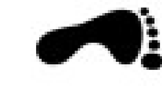
26 de enero del 2009



9 de marzo del 2009



Marcamos Huella

http://atix.opentelematics.org